

CompuServe Interactive Services

# **Computer Room Operations TRAINING MANUAL**

**Last Update: 1/5/2000 (ME)**

**File Name: TRAINMAN.DOC**



## **INTRODUCTION**

**OPERATIONS' MISSION:** Provide the resources and support necessary to maintain the highest degree of customer satisfaction, while continuing to allow for both company and personal growth.

Due to shift work, an operator is exposed to different hours, equipment, and tasks in Operations. As a new operator, you will find that it takes time to learn procedures and tasks that you will need to know.

To advance in Operations, an operator must be comfortable with the background material and be ready to learn much more. It is easy to just memorize how to do something, especially when you do the same thing day after day. The difficulty lies in understanding what you are doing, why you are doing it, and how it effects other tasks or procedures.

You will probably not retain all knowledge covered in your initial Operations training. If you remember the basic concepts and know some of the more commonly used programs or procedures, you should do well on the job. As your knowledge of daily tasks and procedures grows, the pieces will come together, and you will realize that you understand.

Your supervisor will try to keep track of where you stand on knowledge and what you need to learn. Of course, ask questions. This cannot be emphasized enough. You will learn about things much quicker by asking questions. Whenever you have the least doubt about what you are doing, always ask!

There are reference manuals in the computer room, your training manual, and other documentation available for learning about Operations. Your immediate supervisor is always your primary trainer.



## BACKGROUND INFORMATION

- Short history of Operations
- Types of services
- Types of customers
- Computer room work
- Manuals and assigned work areas

## OPERATOR INFORMATION

## SCHEDULES AND OVERTIME/HOLIDAY/SICK COMPENSATION

### DAY SHIFT SCHEDULE

Computer Room Supervisor		07:30 – 16:00 (Mon-Fri)	
Normal Hours		Overtime Hours	
NOC OPR	0700 – 1730	NOC OPR	0700 – 1700
Data Center OPR	0700 – 1730	DC OPR	0700 – 1700
Data Center OPR	0700 – 1430 (M-F)	DC OPR	0700 – 1400
Data Center OPR	0800 – 1830	DC OPR	0800 – 1800

### ROTATION (DATA CENTER)

S	S	M	T	W	T	F
X	X	O	O	X	X	O
O	O	X	X	O	O	X

### ROTATION (NOC)

S	S	M	T	W	T	F
X	X	O	O	O	X	X
O	O	X	X	X	O	O

## **AFTERNOON SHIFT SCHEDULE**

**Shift differential: +7% if at least 4 hours worked past 17:00.**

Computer Room Supervisor		12:00 – 20:30 (Mon-Fri)	
Normal Hours		Overtime Hours	
NOC OPR	1200 – 2230	NOC OPR	1200 – 2200
Data Center OPR	1400 – 0030	DC OPR	1400 – 2400
Data Center OPR	1600 – 2330 (M-F)	DC OPR	1600 – 2300

### **ROTATION (DATA CENTER & NOC)**

S	S	M	T	W	T	F
X	X	O	O	X	X	O
O	O	X	X	O	O	X

## **NIGHT SHIFT SCHEDULE**

**Shift differential: +10% if at least 4 hours worked past 00:00.**

Computer Room Supervisor		2200 – 0830 (Mon-Thu)	
Normal Hours		Overtime Hours	
NOC OPR	2200 – 0830	NOC OPR	2200 – 0800
Data Center OPR	1900 – 0530	DC OPR	1900 – 0500
Data Center OPR	2100 – 0730	DC OPR	2100 – 0700
Data Center OPR	2200 – 0830	DC OPR	2200 – 0800
Data Center OPR	0000 – 0730 (M-F)	DC OPR	0000 – 0700

### **ROTATION (DATA CENTER & NOC)**

S	S	M	T	W	T	F
O	O	X	X	X	X	O
O	O	O	O	X	X	X
X	X	O	O	O	O	X
X	X	X	X	O	O	O

## **WEEKEND DIFFERENTIAL**

AOL policy indicates all hourly associates that work both Saturday and Sunday will receive 12% shift differential. If an associate is scheduled to work both days of the weekend and calls in sick or takes a vacation day on one of those days, weekend differential will not be paid for the weekend day worked and the associate will receive normal shift differential.

## **HOLIDAY PAY (See AOL Holidays HR601 for more details)**

Holiday pay will be calculated based on the employee's straight-time pay rate (as of the date of the holiday) times the number of hours the employee would otherwise have worked on that day. Holiday pay is paid for the 3rd shift starting on the eve of the holiday. 1st and 2nd shifts are paid for the actual holiday.

AOL acknowledges 4 Major and 4 Minor Holidays throughout the calendar year. All Data Center / NOC Associates receive 10 hours of Holiday pay for each of the 4 Major Holidays listed below. If the associate is scheduled to work on a Major Holiday, you will be paid 1.5 times the base rate for all hours worked on the holiday, plus Holiday Pay (base rate times normal scheduled hours). Only the actual hours worked will be counted for overtime for that workweek. If you have a scheduled day off that falls on a Major Holiday, you receive Holiday Pay only (base rate times normal scheduled hours). If you find your own coverage and take vacation on a major company holiday, you get paid holiday pay only and vacation time is not charged to the employee.

<b>Worked:</b>	One shift's pay + Time and a half
<b>Day off:</b>	One shift's pay

<b>Major AOL Company Holidays</b>
New Years Day (January 1 <sup>st</sup> )
Independence Day (4 <sup>th</sup> of July)
Thanksgiving Day (Last Thursday of November)
Christmas Day (December 25 <sup>th</sup> )

A personal holiday will be added to your personal holiday account for each of the four Minor Holidays, regardless if you are or are not scheduled to work on the Minor Holiday. The personal holiday hours you receive are equal to the hours of one normal shift (10 hours). If an associate is scheduled to work on a Minor Holiday and calls in sick, the associate will not receive the benefit of this additional time off and will only receive sick time (see also **SICK TIME**).

<b>Worked:</b>	One shift (10 hrs) added to PFD Bank
<b>Day off:</b>	One shift (10 hrs) added to PFD Bank

<b>Minor AOL Company Holidays</b>
Martin Luther King, Jr. Day
Memorial Day
Labor Day
Day after Thanksgiving

## **FLOATING HOLIDAYS**

In addition to the recognized holidays previously listed, eligible full-time associates receive three floating holidays (30 hours total) on January 1<sup>st</sup> of each year. Floating holidays must be used within that year. Unused PFHD hours by Dec 31 will be lost come January 1. The number of floating holidays earned during an associate's calendar year of hire is based on hire date. Full-time associates hired before March 31<sup>st</sup> receive three days (30 hours), those hired between April 1<sup>st</sup> and June 30<sup>th</sup> receive two days (20 hours), those hired between July 1<sup>st</sup> and September 30<sup>th</sup> receive 1 day (10 hours). Associates hired on or after October 1<sup>st</sup> do not receive floating holidays for the calendar year of hire.

## **SHIFT SWAPS**

Shift Swapping is a departmental benefit, which allows associates to work for each other (or "SWAP") on normal scheduled workdays. This is done in order to accommodate one or both parties in the time of hardship or scheduling conflicts without having to deplete the associates vacation or personal holiday bank. To swap shifts, both Operators must send email to [CSCHEDULE@CS.COM](mailto:CSCHEDULE@CS.COM) and [CRS@LISTSERV.CS.COM](mailto:CRS@LISTSERV.CS.COM) or the Senior NOC Analysts. This will serve as an agreement between both parties that each is now responsible for coverage of the other's shift. All shift swaps must occur within a single pay period, should not create normal scheduled overtime, and must be approved by a Computer Room Supervisor / Senior NOC Analyst. If the swap is approved, it will be posted in the work schedules. Be aware that you may have other obligations in addition to the shift swap (i.e. scheduled overtime, etc.). Once approved, each Operator will be paid for the actual day he/she works (e.g. swapping a regular day for a holiday).

## **OVERTIME**

Scheduled overtime is an integral part of the Data Center Operator and Associate NOC Analysts job description and is required. Overtime pay is based on actual hours worked. Time off, including but not limited to holidays, sick leave, vacation leave, paid time off, or any leave of absence, will not be considered hours worked for purposes of performing overtime calculations.

<b>Scheduled:</b>	Time and a half (if over 40 hours in a pay week).
<b>On a Major Holiday:</b>	1 full shift Holiday pay (10hrs) + 2.5x actual time worked (40hrs+) 1 full shift Holiday pay (10hrs) + 1.5x actual time worked (<40hrs)
<b>On a Minor Holiday</b>	Time and a half (if over 40 hours in a pay week).

All overtime shift start times are noted above and are to be 10hrs in length. The associate is permitted to leave at the 7<sup>th</sup> hour mark only if approved by a CRS and the entire following shift has arrived for duty. If the following shift has a "no-show" or a "call-off", the person working the overtime is then required to work the entire 10-hour shift.

Each associate is required to work a minimum of 12 additional shifts per year. Calling off on a scheduled overtime will not be tolerated. If an associate calls off on a scheduled overtime without a written doctor's excuse 3 times within one year (timer starts with the 1<sup>st</sup> call off), the following will occur:

- 1<sup>st</sup> Occurrence - Verbal warning.
- 2<sup>nd</sup> Occurrence - Written warning.
- 3<sup>rd</sup> Occurrence – Could result in termination.

Associates are permitted to sign up for additional overtime, if so desired. Be aware that you may have other obligations in addition to this additional overtime (i.e. pre-scheduled overtime, classes, etc.). The NOC staff is eligible to sign up for Data Center overtime voluntarily but will not be assigned overtime in the Data Center. The NOC staff can also be called as a last resort when looking for Data Center coverage if an associate calls in sick. Data Center Operators are not eligible to work NOC scheduled or short-notice overtime at this time.

## **SICK TIME**

Up to five days (per occurrence) due to illness, injury or doctor visits. Sick days are NOT to be used as personal days, but may be used for a sick child, parent or spouse (up to 10 days per year). A doctor's excuse is mandatory if the associate misses more than 3 consecutive days. A doctor's excuse may also be required if suspicion of sick time abuse occurs. Before returning to work from a sick leave absence of 5 calendar days or more, an employee must provide a physician's verification that he or she may safely return to work. If an employee calls off the day following or prior to a company Holiday without a physicians note, sick time will not be paid (NPL).

Abuse of sick time will be dealt with accordingly as follows:

1. Written warning on the first occurrence. If this abuse happens in the future, you will be required to provide us with a physician's excuse. The written warning may be skipped if blatant abuse is observed and you may not be compensated for the time taken.
2. If after the first warning, a second occurrence happens and a physician's excuse is not provided, there will be no compensation for that time taken (NPL), and further disciplinary action may be taken.

Examples of suspected abuse in sick time would include, but not limited to: patterns; excessive amounts (without an excuse); at prime/convenient times (Prior to or immediately following a vacation or swap); and calling in sick on a day you have previously attempted to take as a vacation and been denied.

## **ATTENDANCE**

Regular attendance during all scheduled hours of work is essential for the efficient performance and growth of each CompuServe Interactive Services associate. Unsatisfactory attendance, including frequently reporting late or leaving early, may be cause for disciplinary action. If an associate is unable to report for work or will be late for work, he/she must page the CRS as far in advance as possible. The extra time is needed to find coverage for the shift.

## **VACATION TIME**

1 week of vacation = 35 hours. 1-3 years = 70hrs / 4-6 years = 105hrs / 7+ years = 140hrs. If you find your own coverage and take vacation on a company holiday, you get paid holiday pay only and vacation time is not charged to the employee.

## **VACATION GUIDELINES**

1. Once the schedules come out you are responsible for covering your scheduled OT. The schedule will be locked and posted 30 days in advance.
2. Vacation requests must be submitted at least 30 days in advance. Less than 30 days notice will leave the Operator accountable for finding his/her own coverage.
3. Only 2 operators can be on vacation at a time per day. An Operator may arrange his/her own coverage if desired days off are not available. CRS/Sr. NOC Analyst approval is needed in this situation.
4. An Operator can arrange for his/her own coverage if he/she wants to use vacation on holidays and the involved dates below:

Martin Luther King, Jr. Day – 2 days prior through 2 days following actual Holiday
Memorial Day Weekend – 2 days prior through 2 days following actual Holiday
Fourth of July – July 2nd through July 6th
Labor day – 2 days prior through 2 days following actual Holiday
Thanksgiving – Tuesday prior to through Sunday following
Christmas & New Years – December 23rd through December 27 <sup>th</sup>
New Years Day – December 30 <sup>th</sup> through January 3rd

5. If taking a vacation over a holiday or with less than 30 days notice, the scheduler will require an email message stating who will be working for whom and an email message from the person covering as

confirmation. If the scheduler only receives a request from the person wanting the day off, does not get a confirmation note from the person who has agreed to work and no one shows up to work that shift, the shift coverage responsibility falls back to the person originally scheduled to work. Once the scheduler has a confirmation note, the person who has agreed to work is responsible for covering that shift. Should the person who has confirmed to work a shift cease to work in the computer room, the person originally scheduled assumes the responsibility. Please remember that this will be in addition to normal and other OT you may be scheduled to work.

6. In the above where applicable, meaning scheduled OT & short notice vacation and holidays, NOC operators are to work for NOC operators, Data Center Operators are to work for Data Center Operators except in emergencies and/or approved by a CRS/Sr. NOC Analyst. NOC Operators may be called to cover Data Center shifts for short-term coverage.
7. A maximum of 2 weeks (70 vacation hours) will be covered at a time. (If needed, you may swap shifts to extend a vacation longer than 2 weeks). You can also find your own coverage to extend a vacation beyond 2 weeks. If you do this you will need to cover either the first few days or the last few days. The scheduler will only schedule coverage for 2 consecutive weeks of time off.
8. During the months of June, July and August a maximum of 2 weeks or 70 hours of vacation may be taken. You can submit requests for more vacation. However, if you have already taken or have allocated your 70 hours of vacation for June, July and August and another person who has not allocated or used vacation time during this 3 month period submits a request for the same period you requested off (above your original 70 hours), the second person will get the time off and that week may be blocked out. You may still have the time off provided you find your own coverage. Please submit your vacation requests in order of priority for this 3-month period. This is an effort to try and be fair to everyone who would like some summer vacation time to enjoy.
9. When working holidays, if you decide to split the shift overlap (this is acceptable), you must put the hours you actually worked on your time sheet. The time sheet is a legal document and should always reflect the true hours worked. You will be paid 10 hours Holiday pay for the day plus 1.5x for the hours worked. The sum of these two totals equals 2.5x for the hours worked. If you wish to use vacation time on a scheduled company Major Holiday, you will be compensated Holiday pay only and vacation time will not be deducted. To get the entire holiday off in this manner, you must either find your own coverage or have the approval of the involved CRS's/Sr. NOC Analysts stating they do not mind working short handed.
10. If a CRS/Sr. NOC Analyst allows one of his people to take time off without coverage, then they will need to get the approval of the other CRS's/Sr. NOC Analysts working that day. Another CRS or Sr. NOC Analyst might have given one of his people that day off too. This will insure all centers and shifts will have adequate coverage.

**NOTES:**

- Make sure you send an email message for ALL vacation requests to **INTERNET:CSCHEDULE@CS.COM** and your CRS/Sr. NOC Analyst. Short notice vacation requests or swaps will be added to the schedule once confirmation notes from both parties have been received.
- Please check for updates on the schedule.
- Please keep in mind that any questions or problems with the schedules should be directed to your CRS/Sr. NOC Analyst. He will work with the scheduling software to resolve any issues.

**AOL's Appearance Policy**

HR Policy 730

Personal Appearance

AOL is proud of its relaxed atmosphere; casual dress is the norm. Employees should wear whatever they feel is most conducive for them - and their coworkers - to get the job done. At the same time, employees should take care to dress according to the requirements of their positions.

Attire should be clean and neat but not overly revealing, offensive or otherwise distracting. Also, general rules of personal hygiene apply. Employees should be aware that strong fragrances, natural or chemical, might be offensive to coworkers in a close environment.

Sensitivity to these conditions goes a long way toward making a more productive work environment. If an employee's inappropriate dress affects work quality or productivity, management may send the individual home to change on personal time without pay.

Status: Approved

Effective Date:

Transmitted: 2/25/98 11:12 AM

### ***CompuServe Interactive Services Operations' Guidelines***

#### **All shifts and All times:**

- No crop tops, cutoffs, or sandals.

#### **08:00 - 17:00 Monday through Friday:**

- T-shirts and collared shirts are acceptable as long as they have no offensive slogans on them.
- Denim jeans of any color are acceptable as long as they are conservative and clean.
- Tennis shoes may be worn as long as they're clean.
- *Time sheets*
- *Meetings*
- *Operations positions*
- *CSIMEN online*

## **SHUTTLE RUNS**

At times it is necessary to transfer magtapes and other data between computer centers, especially for files that are too large to send via the network. At key times of the day the CRS may request that an Operator run a "shuttle" to the opposite center to take anything being sent and bring back anything being sent from the opposite center. The items being sent are usually found in the "SHUTTLE" basket in either center.

Transportation for shuttle runs should be in a company vehicle whenever possible. Operators may use their own cars only if a company car is unavailable. In this case, the Operator is reimbursed for mileage on a monthly basis.

**NOTE:** Any one that drives a company car must sign a form that allows CompuServe to check the person's driving record. See the Management Handbook for restrictions.

Please submit an EXPENSE REPORT for mileage accrued during the past month for reimbursement.

## **SECURITY**

- *Important aspect of job*
- *Building and computer room access*
- *Different badge colors*
- *Escort badges for tours*
- *Spare badges*
- *Confidential data*
- *Separation of users data is guaranteed*
- *Password security*
- *Computer room passwords changed monthly*
- *Listing are shredded instead of thrown away*
- *Recycle only non-confidential paper*
- *Must know last month's password to give passwords over the phone*
- *Local security: 538-4394 (Arlington office)*
- *email: aolhelp4u@aol.com*

## SYSTEM CONFIGURATION

### Processors

MAINFRAMES	
AH?	VAX (Accounting cluster)
BH?	CompuServe host - Arlington
CH?	CompuServe host - Arlington
DH?	CompuServe host - Dublin
EH?	CompuServe host - Dublin
FH?	CompuServe host - Dublin
GH?	CompuServe host - Arlington
HH?	CompuServe host - Dublin
IH?	CompuServe host - Dublin
JH?	CompuServe Y2K host - Dublin
KH?	CompuServe Y2K host - Dublin
MICROHOSTS	
HPA??	HP Unix - Arlington
HPD??	HP Unix - Dublin
HPH??	HP Unix - Hilliard
MHA??	Microhost - Arlington
MHD??	Microhost - Dublin
QH???	Microquote
NT???	Windows NT machines
???-GW-#	Arlington, Dublin and Hilliard Inbound Gateway machines
???-IMG-#	Arlington, Dublin and Hilliard Internet Mail Gateway machines

### NON-PROD2 NT HOST NAMING CONVENTION

Each host contains an eight character name. The following is a breakdown of NT host, NTAPBAAA:

Character Position	Host Name	Description
1	N	The first two positions represent that the machine you are dealing with is a NT machine.
2	T	
3	A	Data Center Location: A = Arlington, D = Dublin, H = Hilliard
4	P	Domain the server is located in: P = Production, T = Test, N = Notes, C = CompuServe, I = Inhouse and W = Web, D = Development*
5	B	%Type of machine: F = Front-End, B = Back-End, S = Server, D = Domain Controller, W = Web Server
6, 7, 8	AAA	Unique three character identification string (i.e. AAA, AAB,

\* Development is not actually a domain, but a special setup with the developers.

%	Type of machine information.
<b>B</b>	Back End, Gateway, or File Servers not in the INHOUSE domain. <ul style="list-style-type: none"> <li>Servers without X.25 connections and not a web server.</li> <li>Notes servers have X.25, but still get the Back End name.</li> </ul>
<b>S</b>	File Server <ul style="list-style-type: none"> <li>This letter is only used in the INHOUSE domain for file servers.</li> <li>This letter also represents the SMS SQL server in the CompuServe domain.</li> </ul>
<b>F</b>	Front End Server <ul style="list-style-type: none"> <li>Servers with X.25 connections.</li> </ul>
<b>D</b>	Domain Controller
<b>W</b>	Web Server

### PROD2 NT NAMING CONVENTION

Each host contains an eight character name and the following is a breakdown of an NT host, NFH2FAAA:

Character Position	Host Name	Description
1	N	<b>Operating System:</b> N for NT, O for OS2, U for Unix
2	F	<b>Product:</b> F = Forum, M = Mail, O = Our World, X = Operations
3	H	<b>Data Center Location:</b> A = Arlington, D = Dublin, H = Hilliard
4	2	<b>NT Domain:</b> 2 = Prod2, 3 = Prod3
5	F	<b>Type:</b> F = Front-End, B = Back-End, X = Service Machine, Q = SQL Server, W = Web Server
6, 7, 8	AAA	<b>ID:</b> Unique 3 character identification string

### DISK DRIVES

DEVICE NAMES	
DZ??:	SCSI disk on mainframe
DU????:	VAX disk drive
PRIMARY STRUCTURE NAMES	
BK?:	Primary structure for BH?
DK?:	Primary structure for CH?
DD?:	Primary structure for DH?
ED?:	Primary structure for EH?
DF?:	Primary structure for FH?
GK?:	Primary structure for GH?
HD?:	Primary structure for HH?
DI?:	Primary structure for IH?
JK?:	Primary structure for JH?

<b>SECONDARY STRUCTURE NAMES AND TYPES</b>	
<b>I??:</b>	Secondary structure - CIS customers and products
<b>Z??:</b>	Secondary structure - CompuServe Mail (IPX)
<b>FD?:</b>	Secondary structure - Financial customers and products
<b>MS?:</b>	Secondary structure - MIS Department
<b>AC?:</b>	Secondary structure - Accounting Department
<b>???:</b>	Other Secondary Structures
<b>????:</b>	Mountable structures
<b>T???:</b>	Test structures
1234	4 = ( LETTER OF ACTUAL TEST STRUCTURE IE. A..B..C..) 3 = ( LAST LETTER OF HOST NAME ) 2 = ( FIRST LETTER OF HOST NAME )

## DATA STORAGE

<b>CompuServe 36 bit hosts</b>	
<b>WORD</b>	36 bits or 5 DEC ASCII characters (7 bits) + 1 parity bit
<b>BLOCK</b>	640 characters or 128 words
<b>CLUSTER</b>	5 blocks (For CIS structures) or 5 blocks (For primary structures and most customers)
<b>NOTE:</b> Cluster size varies from 5 blocks to 23 or more blocks, depending upon the type of users on a structure. When the structure is created, the cluster size is set with ONCE-ONLY or ZWIMAL. The Monitor program (Disk Operating System) allocates disk storage by cluster, so if a file uses 1 block of a 5-block cluster, there are still 4 unused blocks assigned to the file that are wasted. In general, users that create large files have a larger cluster size so that files can be Copied, etc. faster. Users that normally create many smaller files have a smaller cluster size to prevent wasted blocks.	
<b>PAGE</b>	512 words or 4 blocks
<b>K (Kilo-word)</b>	2 pages or 8 blocks or 1024 words
<b>VAXES</b>	
<b>BYTES</b>	8 bits
<b>WORD</b>	16 bits or 2 bytes
<b>LONG WORD</b>	32 bits or 4 bytes
<b>QUADWORD</b>	64 bits or 8 bytes
<b>OCTAWORD</b>	128 bits or 16 bytes
<b>BLOCK</b>	512 bytes or 256 words or 128 long words
<b>PAGE</b>	512 bytes or 1 block
<b>CompuServe 32 BIT HOSTS</b>	
<b>BYTE</b>	8 bits
<b>KILOBYTE</b>	1024 bytes (one thousand bytes)
<b>MEGABYTE</b>	1024 kilobytes (one million bytes) or 1,048,576 bytes
<b>GIGABYTE</b>	1024 megabytes (one billion bytes)



## Review

1. Job information
2. Work schedules
3. Manuals
4. Time sheets
5. Security Guidelines
6. Types of hardware
7. Host and structure names
8. Device names

# Lunch

# Computer Room Tours

### **36 BIT CUSTOMER IDENTIFICATION**

- Project Programmer Number (PPN)
- Maximum of 6 project, 6 programmer numbers
- All PPNs are octal ( base of 8 ( base of 10 is decimal))
- Important PPNs for operations

<b>1,2</b>	System Administration
<b>100,4</b>	Tapes / Tasks System
<b>131,1</b>	Computer Room Operations work area.

- Powers of [1,2]
- Difference between inhouse and customer PPNs

### **NT ACCOUNTS**

- Username (up to 15 characters) and password
- Operations Accounts
  - administrator
  - backup
  - personal

### **BSDI ACCOUNTS**

- Username (up to 8 characters) and password
- Operations Accounts
  - root
  - shutdown
  - opr

## **LOGGING ONTO A HOST**

- What is a UPF file?
- What is a LOCATE file?
- How to get USER ID: prompt
- Logging on using the @
- How to log on using an ACCOUNT ID
- Difference between logging onto a switchable and a CTY
- To log-off, OFF HELLO BYE LOG
- Logging onto BSDI via CTY, LAN and X.25
- Logging onto NT
- Logging onto VAX via CTY or switchable

## **YOYOED**

- Used to edit the locate (yoyo) file DSK:[1,201]LOCATE.PFE
- LOCATE.PFE contains structure defaults for all PPNs

## **REACT**

- Used to generate and edit User Parameter Files
- Defines all privileges and restrictions
- UPF file structure
- System administrator level
- Project administrator level
- Look at examples of .UPF files

### **UPF file Example (project 140)**

```
Pname:  OPERATIONS
        Priv:   NEA CPW NWU LTI
        Core:   96.0P,96.0P  (96.0,96.0)
        Locate: LPT:COL
        Level:  1
        Route:  Operations
                Please deliver to CCR Thank-you
        Expiration date: 9-Jun-99 00:00:00
        Class:  .INH
        Quota:   ALL:    10000
        FILPRO:  (4)
        UFDPRO:  <740>
        JSL:     CAF,*,SSL/QUIET/NOUFD
        Library: [140,140]
        Pwd Chg Freq:    370 (370)
1      ** Programmer **
        Name:    TRAINING
        Priv:    REA CLT SDS LBL CLR
        Logo:    the school of hard knocks
        Route:   Operations Training
                Dennis Noland
```

```
2      ** Programmer **
      Name:  DAVE KARR
      Priv:  CLT LBL CLR
3      ** Programmer **
      Name:  MELVIN PAYNE
      Priv:  CLT LBL CLR
      Logo:  VULCAN SPACE CENTRAL
      Route: MEL PAYNE
           BUILDING 1  Quota:  ALL:20000
1      ** Line **
      A.I.D.: MOVIE
      Priv:  CLT LBL CLR
      DO:    CAM:MOVIE.COM[140,3]
2      ** Line **
      A.I.D.: PHONE
      DO:    CAM:PHONE.COM[140,3]
      Priv:  CLT LBL CLR
```

## **NT**

- User Profile under Administrative Tools

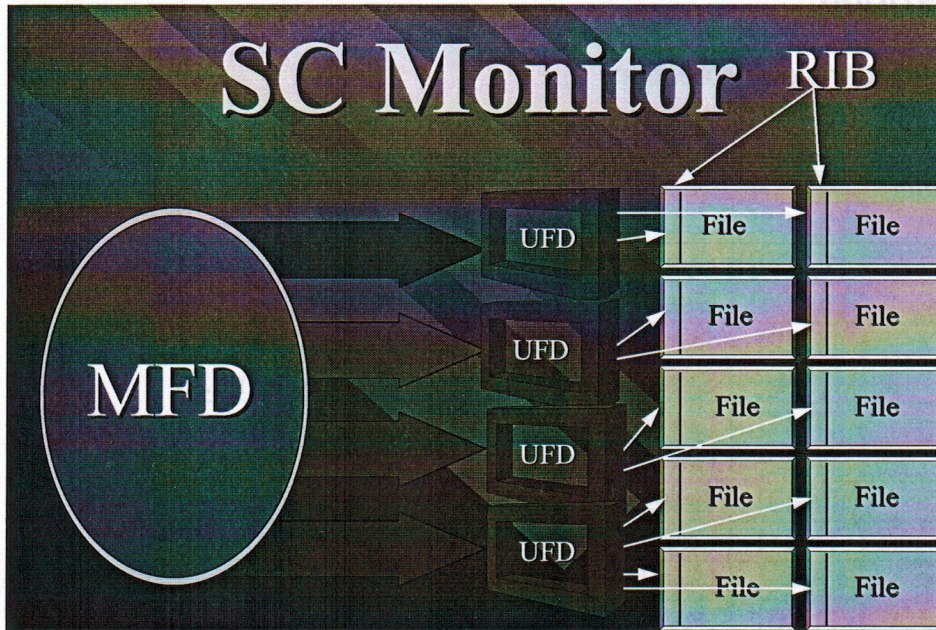
## **BSDI**

- Password file

## **DISK OPERATING SYSTEMS**

- SC The Monitor (MONITR.EXE)
- BSDI The Kernel (1.1, 2.1)
- Windows NT (3.51, 4.0)
- OS/2
- HP-UX The Kernel (10.10, 10.20)
- Different versions of all Operating Systems

## SC MONITOR



- Master File Directory (MFD)
  - Tells Monitor where to find the UFD.
  - A refresh deletes the MFD for the structure.
- User File Directory (UFD)
  - Contains a list of PPN's file names and RIB pointers.
  - A CATalogue command lists a UFD.
  - [1,1]#140,,1.UFD
  - UFD for a PPN exists only if there are files for the PPN.
- Retrieval Information Block (RIB)
  - Leading block of each file.
  - Contains cluster pointers and detailed data on each file.
  - A DIRectory command lists part of the RIB.
- Storage Allocation Table (SAT)
  - Used by the Monitor to keep track of clusters free / in use.
  - [1,4]SAT.SYS
  - DSKRAT uses the SAT when examining clusters for a disk.
- System software areas [1,\*]

## **FILENAME SPECIFICATIONS**

- device:[p,pn]filename.extension(protection)
- Pseudo-devices
- Examples:

<b>Pseudo Device Name</b>	<b>PPN</b>	<b>Description</b>
<b>LIB:</b>	[1,103]	.COM file library
<b>BLG:</b>	[1,200]	System billing area.
<b>UPD:</b>	[1,201]	User Parameter File area
<b>SPT:</b>	[1,22]	System support area.
<b>SYS:</b>	[1,4]	System area for common programs.
<b>LV1:</b>	[1,41]	Level 1 program area.
<b>LV2:</b>	[1,42]	Level 2 program area.
<b>LV3:</b>	[1,43]	Level 3 program area.
<b>LV4:</b>	[1,44]	Level 4 program area.
<b>LV5:</b>	[1,45]	Level 5 program area.
<b>LV6:</b>	[1,46]	Level 6 program area.
<b>DSK:</b>		Default disk structure.
<b>ALL:</b>		Search on all structures.
<b>TTY:</b>		Your terminal.

- Filename - Up to 6 alphanumeric characters
- File extension - Up to 3 alphanumeric characters
- A period separates the filename from the extension.
- First character of name or extension should be alphabetic
- Importance of extensions
- Wildcard characters - ? and \*
- Protection levels
- Monitor assumptions for omitted fields (defaults)
  - Device - DSK:
  - PPN - the one you are logged into
  - Filename - None (sometimes \*.\* )
  - Protection - (3) or whatever is set up in the UPF
- BSDI file specs.
  - Case sensitive
  - Up to 14 characters
  - No imbedded blanks
  - Avoid special characters, except: \_ .
  - Pathname: /directory/subdir/filename

- NT file specs.

- Filename.Extension: Filename 8 characters, Extension 3 characters.

**NOTE:** NT supports long filenames much larger than 8 characters, but by using 8.3 as a standard, file names do not get truncated when moved to a DOS based OS.

- Not case sensitive, but keeps case.



# Review

1. Customer identification
  - a) PPNs
  - b) Powers of [1,2]
2. Logging on the system
  - a) UPF files
  - b) Getting the USER ID: prompt
  - c) Logging on @STRUCTURE
  - d) Using Account ID's
3. REACT
  - a) Modifying .UPF files using REACT
  - b) Different command levels
4. YOYOED
  - a) Modifying LOCATE.PFE using YOYOED
  - b) LOCATE.PFE contains default structures for all PPNs
5. Online Storage
  - a) MFDs, UFDs, RIBs
  - b) [1,\*]
6. Filename specifications

## **COMMAND SYNTAX**

Terminating a line

- Carriage-return <cr>
- Escape <esc> <\$>

Command syntax

- COMMAND
- ARGUMENT
- SUBCOMMAND
- /SWITCH

Difference between Monitor Mode and Program Mode

ICS help facility

## **CONTROL CHARACTERS**

<b>^C</b>	Halt program run, when logging in, will connect to network and prompt for USER ID:
<b>^D</b>	*** <b>WARNING</b> *** Disconnect job from system
<b>^O</b>	Interrupt output but will not halt program run
<b>^P</b>	Used by some programs as an interrupt
<b>^R</b>	Retype line just typed in, CTYs only
<b>^U</b>	Cause program to ignore data typed in and return line pointer to beginning
<b>^V</b>	Works like ^R, switchable terminals only

## **COMMON ICS PROGRAMS**

### **COPIER**

- Used to copy files between two areas on disk
- Invoked by commands COPY, RENAME, or COMBINE
- Examples:

<b>COPY</b>	COP STR: [P,PN] FILE.DAT TO STR: [P,PN] FILE2.DAT
<b>COPY/Remote</b>	<div><div>COPY STR: [P,PN] FILE.EXT TO STR: [P,PN] *.* /R COPY STR: [P,PN] *.* FROM STR: [P,PN] FILE.EXT /R <b>Example:</b> <div>COP DKF: [1,2] FILE.DAT TO DKB: [131,1] *.* /REM  Password for DKB: [131,1] on CHB ?&lt;enter password&gt; Account ID for remote (or &lt;CR&gt; for none) ?OPR  Transfer started on DKF:FILE.DAT[1,2]  Blocks transferred:        1 (100%)  Transfer Complete!  OK</div></div></div> <div><b>NOTE: YOU WILL BE PROMPTED FOR PASSWORD AND ACCOUNT ID(IF ANY).</b></div>
<b>RENAME</b>	REN STR: [P,PN] FILE1.EXT TO STR: [P,PN] FILE2.EXT
<b>COMBINE</b>	COMBINE ONE.DAT,TWO.DAT TO THREE.DAT

### **DIRECT**

- Used to read directory of users files and list specified information.
- Invoked by commands CATALOG and DIRECT
- **CAT**alog (listing of UFD)
- Example:

<b>CAT</b>	CAT STR: [P,PN] FILE.DAT
------------	--------------------------

- **DIRectory**

Switch	
<b>/A</b>	ACTUAL (AS OPPOSED TO ALLOCATED)
<b>/B</b>	BLOCKS
<b>/BIN</b>	BINARY
<b>/G</b>	DISPLAYS A GRAND TOTAL OF STORAGE
<b>/L</b>	LONG DIRECTORY
<b>/Q</b>	LISTS DISK QUOTA FOR A [P,PN] (DISK STORAGE LIMIT)
<b>/SOR:OPTION</b>	SORT BY...
	<b>EXT</b> EXTENSION OF FILE
	<b>CRE</b> CREATION DATE OF FILE
	<b>ACC</b> LAST ACCESS DATE OF FILE
	<b>NAM</b> FILE NAME
	<b>PRO</b> PROTECTION CODE
	<b>SIZ</b> SIZE:ASC (ASCENDING ORDER) :DES (DESCENDING)
<b>/T</b>	DISPLAYS TOTAL STORAGE FOR A [P,PN]

- Example:

<b>DIR</b>	DIR/A/B/L STR: [P,PN] FILE.DAT
------------	--------------------------------

### **LISTER**

- Used to provide listed output
- Invoked by commands TYPE, or DELETE
- Example of typing out a specified file(s) without a heading:

<b>TYPE FILE</b>	TYPE FILE.DAT
------------------	---------------

- Examples of deleting a file from disk:

<b>DELeTe</b>	DEL NAME.EXT DEL DKE: [131,1] *.DAT
---------------	--

## **CSNPAK**

- Used to pack several specified files into one master file
- Invoked by command PACk
- Gives extension of .PAK
- Examples:

<b>PAC COMMANDS</b>	<b>Description</b>
PAC ADD FILE.EXT TO MASTER.pak	ADDS FILE.EXT TO MASTER.PAK.
PAC CAT MASTER.pak	CATALOG OF FILES WITHIN MASTER.PAK.
PAC CLean MASTER.pak	DELETES ANY DISK FILES THAT HAVE THE SAME SPECS AS FILES IN MASTER.PAK.
PAC DIR/sw MASTER.pak	DIRECTORY OF FILES WITHIN MASTER.PAK VALID DIRECTORY SWITCHES CAN BE USED.
PAC MAKE MASTER.pak FROM FILE1.EXT, FILE2.EXT...	CREATES MASTER.PAK FROM DISK FILES.
PAC RETrieve FILE FROM MASTER.pak	PLACES A COPY OF SPECIFIED FILE FROM MASTER.PAK TO YOUR DISK AREA.
PAC PURge FILE.EXT FROM MASTER.pak	DELETES SPECIFIED FILE FROM MASTER.PAK
PAC TYPE FILE FROM MASTER.pak	TYPE A FILE WITHIN MASTER.PAK
PAC UPDate MASTER.pak	OVERWRITES ANY FILES IN MASTER.PAK WITH DISK FILES THAT HAVE THE SAME SPECIFICATIONS.

## **VI EDITOR**

### **SYNTAX**

vi <filename> (opens file)

### **COMMAND MODE COMMANDS**

<b>SCROLLING:</b>	
<b>^E</b>	<b>Down one line</b>
<b>^Y</b>	<b>Up one line</b>
<b>^D</b>	<b>Down 1/2 page</b>
<b>^U</b>	<b>Up 1/2 page</b>
<b>^F</b>	<b>Forward 1 page</b>
<b>^B</b>	<b>Back 1 page</b>

### **ex mode commands**

syntax:        :x,yCommand where x is the first line # of range and y is the second (optional) line # of range. \$ is shorthand for the last line of the file. ex commands must be followed by a carriage return.

s/OLD/NEW/g	Substitute NEW for OLD (OLD may be a regular expression). Optional g replaces all occurrences of OLD on each specified line; otherwise, only the first occurrence is replaced.
!command	Execute UNIX shell command.
w <filename>	Write file.
w!	Write file; overwrite existing file without prompting.
wq	Write file and quit.
q	Quit vi.
q!	Quit vi even if buffer changed. (abandon edit)

### **EDITOR SETTINGS**

syntax:        :set label or :set label=value  
examples:     :set scroll=3 set number  
note:         prepending no to a label turns it off  
example:      :set nonumber

label	What it does
ignorecase	Ignore upper/lower case in searches & substitutions [ <b>ic</b> ]
number	Display line numbers on screen (not in file) [ <b>nu</b> ]
showmode	Indicate text entry mode
tabstop	Number of spaces for tab [ <b>ts</b> ]
terse	Make error and info messages briefer (!)

**NOTE: Abbreviations are in square brackets following descriptions.**

REGULAR EXPRESSIONS	
May be used in / (search) commands, or in s// (substitute) commands.	
.	Match 1 character
*	Match 0 or more of preceding 1-character RE
+	Match 1 or more of preceding 1-character RE
^	Match beginning of line
\$	Match end of line
\<	Match beginning of word
\>	Match end of word
[xyz]	Match any 1 of characters x, y, or z
[^ab]	Match any 1 character not a or b
\	Turn off special meaning of characters

### **ex mode commands**

syntax:        :x,yCommand where x is the first line # of range and y is the second (optional) line # of range. \$ is shorthand for the last line of the file. ex commands must be followed by a carriage return.

s/OLD/NEW/g	Substitute NEW for OLD (OLD may be a regular expression). Optional g replaces all occurrences of OLD on each specified line; otherwise, only the first occurrence is replaced.
!command	Execute UNIX shell command.
w <filename>	Write file.
w!	Write file; overwrite existing file without prompting.
wq	Write file and quit.
q	Quit vi.
q!	Quit vi even if buffer changed. (abandon edit)

### **EDITOR SETTINGS**

syntax:        :set label or :set label=value  
examples:     :set scroll=3 set number  
note:         prepending no to a label turns it off  
example:      :set nonumber

label	What it does
ignorecase	Ignore upper/lower case in searches & substitutions [ <b>ic</b> ]
number	Display line numbers on screen (not in file) [ <b>nu</b> ]
showmode	Indicate text entry mode
tabstop	Number of spaces for tab [ <b>ts</b> ]
terse	Make error and info messages briefer (!)

**NOTE: Abbreviations are in square brackets following descriptions.**

<b>REGULAR EXPRESSIONS</b>	
<i>May be used in / (search) commands, or in s// (substitute) commands.</i>	
.	Match 1 character
*	Match 0 or more of preceding 1-character RE
+	Match 1 or more of preceding 1-character RE
^	Match beginning of line
\$	Match end of line
\<	Match beginning of word
\>	Match end of word
[xyz]	Match any 1 of characters x, y, or z
[^ab]	Match any 1 character not a or b
\	Turn off special meaning of characters

## **GLOBAL CHANGES**

When wanting a global change, use the following command syntax:

**:1, \$s/old string/new string/g** This will start on line one in the file you are editing and change all occurrences of old string to new string. The /g stands for global change.

**:%s/old string/new string/g** This will start from the line that is currently being pointed to and changes all occurrences of old string to new string to the bottom of the file.

## **EMACS EDITOR**

### **SYNTAX**

**emacs <file.ext>** (opens file)

### **COMMANDS**

<b>^f</b>	Forward in the same line
<b>^b</b>	Backwards in the same line
<b>^n</b>	Move to next line
<b>^d</b>	Delete character to the right of the cursor
<b>^p</b>	Move to the previous line
<b>&lt;esc&gt; (then) shift &lt;</b>	Move to the top of the file
<b>&lt;esc&gt; (then) shift &gt;</b>	Move to the bottom of the file
<b>^l</b>	Refresh the screen
<b>^k</b>	Deletes the entire line the cursor is on
<b>^g</b>	Quits current command
<b>&lt;esc&gt; (then) x (then) replace string</b>	Global changes
<b>^s</b>	Search for a string
<b>^xs</b>	Save file
<b>^xc</b>	Exits emacs editor

## **FILGE**

- Used to generate and edit files
- Invoked by commands EDIT, CREATE , FILGE
- Meaning of line pointer
- Editing subcommands

The program filge maintains an invisible "line pointer" that indicates the current line of the message you are editing. Some commands confine their actions to the current line; others begin on the current line and proceed upward or downward in the workspace. Most commands may be specified as a repeated command by enclosing the command itself in parenthesis.

### **Editing Switches**

<b>A</b>	Appends a string to the end of the current line. <b>/A/string</b>
<b>B</b>	Sets the pointer to the last line (Bottom) of the workspace. <b>/B</b>
<b>C</b>	Changes the first occurrence of "string1" in the current line to "string2". <b>/C/string1/string2</b>
<b>D</b>	Deletes "n" lines starting at the pointer. <b>/D n</b>
<b>EX</b>	Exits from edit mode. <b>/EX</b>
<b>GET</b>	Inserts the contents of the alternate workspace into the primary workspace, immediately following the current line. <b>/GET</b>
<b>L</b>	Locates a string. <b>/L/string</b>
<b>MOVE</b>	Clears the alternate workspace and moves "n" lines from the primary workspace to the alternate workspace, starting with the current line. <b>/MOVE n</b>  Moves the pointer to the first line that contains "string", searching from the next line after the pointer to the bottom of the workspace.
<b>P</b>	Prints "n" lines starting at the pointer. <b>/Pn</b>
<b>Q</b>	Quits without saving the current changes. <b>/QUIT</b>
<b>R</b>	Replaces the current line with "string". <b>/R/string</b>
<b>T</b>	Sets the pointer in front of the first line (Top) of the workspace. <b>/T</b>
<b>TCNT</b>	Moves the pointer to the top of the workspace and prints a count of the lines in the workspace, the number of characters used, and the number of characters available. <b>/TCNT</b>
<b>TYPE</b>	Types everything in your workspace, starting at the top. <b>/TYPE</b>
<b>UNC</b>	Unchanges the last change you executed. <b>/UNC</b>

## UTILITY PROGRAMS

### IHMAIN

- Used to get an immediate status of any host or structure.
- Finds host that structures are mounted on or all structures for a host.
- IHMAIN commands:

<b>FIND ISK</b>	TELLS WHAT HOST ISK IS MOUNTED ON.
<b>STR ISK</b>	TELLS WHAT HOST ISK IS MOUNTED ON.
<b>STR IS?</b>	TELLS WHAT HOSTS ALL IS? STRUCTURES ARE MOUNTED ON.
<b>HOST EHA</b>	TELLS WHAT STRUCTURES ARE MOUNTED ON SYSTEM EHA.
<b>STR:</b>	LOGS ONTO THE SYSTEM WITH THAT STRUCTURE ONLINE.
<b>CHA.</b>	LOGS ONTO THE SYSTEM FROM THE SYSTEM WHERE YOU ARE.

- Example:

```
r spt:ihmain

IHMAIN/CHF>find isk
  ISK: is mounted on FHP (OK)
IHMAIN/CHF>str isk

ISK...FHP

Total: 1
* => anomaly
~ => disabled

IHMAIN/CHF>str is?

ISA...EHD  ISW...GHJ
ISB...DHH  ISX...FHH
ISC...CHN  ISY...CHC
ISD...HHN  ISZ...GHE
ISE...CHJ
ISF...DHC  Total: 26
ISG...EHH  * => anomaly
ISH...FHG  ~ => disabled
ISI...FHD
ISJ...CHI
ISK...FHP
ISL...EHL
ISM...CHM
ISN...EHE
ISO...CHU
ISP...DHE
ISQ...CHL
ISR...GHK
ISS...DHL
IST...DHD
ISU...DHW
ISV...FHV
Press RETURN to continue: host eha
  EAA, EDA, FCA, FCB, FCD, FCE, FDA, FDB,
  FDH, FDI, FDK
```

```
IHMAIN/CHF>isk:
IHMAIN/FHP>cha.
IHMAIN/CHA>exit
OK
```

## **EMDUTL**

- Primarily used by programmers to check the delivery of electronic mail.
- Used by Operations to find a structure, project, or PPN.
- For project or PPN, it will list the primary default structure and host, and backup default structure and host (if any).
- Lists free disk storage for a structure.
- EMDUTL commands:

<b>FIND CAF</b>	TELLS WHAT HOST CAF IS MOUNTED ON.
<b>FIND 140,65</b>	TELLS WHAT STRUCTURE AND HOST THE PPN IS LOCATED ON.

- Example:

```
r spt:emdutl

EMDUTL/CHF>fin caf
CAF: is mounted on CHF
EMDUTL/CHF>fin 140,26
[140,26]          CAF    CHF
EMDUTL/CHF>exit

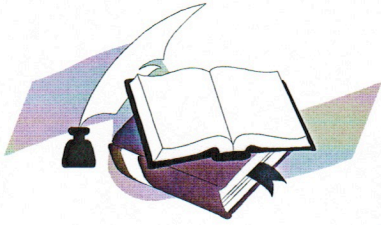
OK
```

## **SNDFIL**

- Used to send a file to CompuServe Mail.
- Need to know the filename to send and the CS Mail address to send to.

```
r sndfil
SNDFIL V 1B(54) 12-May-95 13:13 EDT
Filename: matt.dat
[ Message contains 26 characters ]
Subject: This is where you put your message subject.
Sender's name: Matt
Enter receivers, one per line (<CR> when done)
Send to: INTERNET:MATTEISERT@CS.COM
Send to: <cr>
[ Message sent ]

OK
```



# **Review**

## **1. System configurations**

- Host, structure, and device names
- File specifications
- LOGIN

## **2. Control Characters**

## **3. ICS commands and programs**

- COPIER
- DIRECT
- LISTER
- CSNPAK
- VI
- EMACS
- EDIT(FILGE)

## **4. Utility programs**

- Finding a structure or PPN using IHMAIN or EMDUTL
- Copying files between systems using COPY/REMOTE.
- Sending files to CSMail using SNDFIL



## Examination One

1. Why is [1,2] so special?
2. Name the names of the different Operating Systems we have?
3. What hardware do we use to directly communicate with the processor?
4. What color is LEVEL 3 access badges?
5. When can you give a password over the telephone?
6. Is the PPN [138,6579] valid on our systems? Why or why not?
7. What is a UFD? Explain its use?
8. What is the command to copy the file TABLE.DMS from [22710,22] on CAF: to [131,1] on the primary structure?
9. Explain the use of a .PAK file?
10. What does the COP/REM command allow us to do?
11. What important things will the program IHMAIN allow us to do? EMDUTL?
12. What program is used to send a file to CompuServe Mail?

## **OPSER**

- Difference between switchables and CTYs
- Allows multiple jobs to be run from one terminal
- Subjobs
- Control characters in OPSER

:Xn n = A CONTROL CHARACTER, :XC = ^C.

- Commands

:W	TELLS THE STATUS OF THE SUBJOB OPSER IS POINTED TO.
:W OPR	POINTS TO SUBJOB "OPR" AND GIVES STATUS.
:W ALL	JOB STATUS OF ALL SUBJOBS, POINTER DOES NOT MOVE.
OPR-	POINTS TO SUBJOB "OPR", DOES NOT GIVE STATUS.
WORK-CAT	POINTS TO SUBJOB "WORK" AND PERFORMS THE COMMAND "CAT". DOES NOT GIVE STATUS.
:LOGIN 131,1	LOG A SUBJOB IN FOR [131,1]
:SL	SILENT LOGIN TO [1,2]
:D DWN=	DEFINES THE CURRENT SUBJOB AS "DWN"
:D ZAB=DKB	CHANGES THE SUBJOB NAME "DKB" TO "ZAB" AND MOVES OPSER POINTER TO SUBJOB "ZAB"
OFF, LOG, BYE	LOGS OFF A SUBJOB (MONITOR COMMAND).
:EXIT	EXIT OUT OF OPSER IF NO SUBJOBS ARE ACTIVE.
:HALF	HALF DUPLEX, NO CHARACTERS WILL ECHO TO TERMINAL.
:FULL	FULL DUPLEX, CHARACTER TYPEOUT WILL RESUME.

- OPCON always uses the subjob OPC
- Subjobs OPR and WORK used for Operations
- Auto-jobs

**:A AT1400.ATO[103,700] MANUALLY STARTS AUTO JOB FOR 14:00.**

## **OVERHEAD PROGRAMS**

Overhead programs usually run on every production host. They are programs that perform a particular function that would be too time consuming or impossible for the operating system (the monitor) to run by itself. These programs usually run detached from [1,2].

The program STARTR starts/runs all overhead jobs for a host. When a host is reloaded or when a SYSTEM START command is entered, STARTR reads SPT:AUTSTR.CMD to start any overhead programs for that host.

### **BILDAE**

This is the "billing daemon" program. It receives IPCF billing messages from the "author" programs and updates one of several billing files on disk. BILDAE also gathers network billing from the network nodes.

### **EMDQUE**

This program takes care of delivering electronic mail files to the same host. If the mail goes to another host, it is queued to EMDXFR.

## **EMDXFR**

This program takes care of delivering electronic mail files across hosts.

## **FILDAE**

1. FILDAE is called upon when a job requests to open a UFD. Normally, a UFD is not opened upon log-in.
2. Changes password or terminal parameters on a user's alternate UPF. The program ALTUPD, which is run from the auto jobs, sends UPF changes to FILDAE when FILDAE itself could not make the change (e.g. when a structure is taken down).
3. Updates the last date/time a user logged in.

## **IBSDAE**

Receives inbound statistics information from nodes in the CompuServe network. It only runs on certain hosts that the network nodes know about, and must run in [331,3310].

## **ICON**

This is the program that controls all independent processing on each system. It references the files JOBFIL.ABT & RPTJOB.ABT in [3,7]. ICON also controls the changing of class parameters twice daily for prime/non-prime-time.

## **IHCONN**

The IPCF network connection program, run from [323,340]. This program establishes connections through the network to all other hosts for the program IHXCVR to use.

## **IHXCVR**

This is the program that controls the sending and receiving of interhost IPCF messages. It uses network links established by the IHCONN program. IHXCVR also keeps track of where all structures are mounted. This is used by IHMAIN and LOGIN. IHXCVR also keeps track of all PIDs for interhost IPCFs.

## **OPCON**

The OPERator CONTROL program. This is the operator side of the UCON/OPCON process that controls the MOUNt and DISmount commands to mount disk structures or magtapes. It normally runs as subjob OPC under OPSER on the CTY.

## **OPSER**

OPSER controls the communication of the CTY terminal with 1 to 14 subjobs. When a host is reloaded, there are three subjobs automatically logged in when STARTR processes [1,22]OPR.ATO: OPR and WORK - Used for Operations tasks, and OPC - For running OPCON.

## **SYSDIS**

When running detached from [2,10] on the host, this program performs four basic functions:

1. Performs disk error logging
2. Sends disk error warning messages to the CTY
3. Controls the host crash alarm system
4. Displays job and host stats on a CRT

## **SYSINF**

SYSINF is the Shutdown coordinator when a SYSTEM SHUT is set on a host. It also keeps the PID map in core and assigns PIDs to programs upon request. This is very important for the IPCF functions to work. A PID is a Process ID for the InterProcess Communication Facility.

## **XFRDAE**

XFRDAE is the background file transfer server. It can be used to send files from host to host or move files on the same system. It is driven by IPCFs, so you must use another application (i.e. OPRSEN) to request file transfers.

## **INFORMATION SERVICE**

The following overhead programs are run only when there is an IS?: disk structure on-line.

### **ISDTRK**

This program gathers information on a regular basis from all the jobs logged onto the system in the CIS or EIS project range. It collects this information to a file on disk which is closed out by ISDDAY, which creates a file used by ISDMRG.

### **OPLDAE**

This program runs from [60103,2] on one IS?: structure and collects key statistical information from the DISPLA program when certain marked pages are accessed. OPLDAE stores this information in a file on one IS?: structure. OPLDAY closes out this file daily and renames it, and ACCOPY then collects the files.

### **IPXSPL**

This program is run for each Information Service structure on a host. When running from project 27, IPXSPL is a spooler program for EZPlex on CIS. When running from any other project, IPXSPL is an InfoPlex spooler.

**NOTE:** Any time a Rapid Response Spooler Crash happens, call someone from the Information Technology contact list immediately.

## **STARTR**

- Starts system 'Overhead' jobs running
- Run automatically when system is reloaded
- Invoked by command SYSTEM START
- Uses [1,22]AUTSTR.CMD
- Partial Example:

```
ALL:BATLST,[1,2],,SPT:
ALL:BLMSVR,[1,2],,SPT:
ALL:BILDAE,[1,2],,SPT:,[0,0],(CLO,NED)
ALL:CRSCPY,[1,2],,SPT:,,,,,500
ALL:EMDQUE,[1,2],,SPT:,[0,0],NED,,,,,256
ALL:EMDXFR,[1,2],,SPT:,[0,0],NED,,,,,256
ALL:EMDXF2,[1,2],,SPT:,[0,0],NED,,,,,256
ALL:FILDAE,[1,2],,SPT:,[0,0],CLO,,,,,500
ALL:FTP<FTPD>,[1,2],,SYS:,[0,0],,,,,,512
ALL:WHOCIM,[1,2],,SYS:,[0,0],RRP
ALL:ICON,[1,2],,SPT:,[0,0],NED,,,,,128
ALL:IHXCVR,[1,2],,SPT:,[0,0],NED,,,,,255
ALL:SYSINF,[1,2],,SPT:,[0,0],(RRP,NED)
ALL:WOOF,[1,2],,SPT:,[0,0],CLO
ALL:XFRDAE,[1,2],,SPT:,[0,0],(NED,RRP,CLO),,,,,,511
ALL:CTXSVR,[1,2],,SPT:,[0,0],NED,,,,,256
ALL:IHCONN,[323,340],,DSK:,[323,340],NED
BHA:SYSDIS,[2,10],,SYS:,[0,0],(NED,CLO),,,T42CSE
BHB:SYSDIS,[2,10],,SYS:,[0,0],(NED,CLO),,,T21ATR
BHC:SYSDIS,[2,10],,SYS:,[0,0],(NED,CLO),,,T27ATR
BHD:SYSDIS,[2,10],,SYS:,[0,0],(NED,CLO),,,T13CSE
BHD:IBSDAE,[331,3310],,SPT:,[0,0],(NED,RRP),,,,,,512
BHE:SYSDIS,[2,10],,SYS:,[0,0],(NED,CLO),,,T25CSE
BHF:SYSDIS,[2,10],,SYS:,[0,0],(NED,CLO),,,T23ATR
(This is the first 25 lines in the AUTSTR.CMD file.)
```

- SYSTEM ST.SYS uses [1,22]AUTSTR.SYS to start SYSDIS only.

## **MONITORING AND CONTROLLING SYSTEMS**

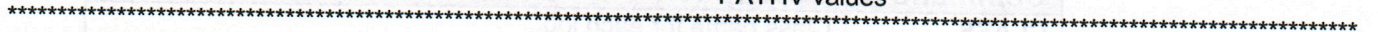
### **SYSDIS**

- Provides a continually updated display of the system status
- Overhead job running on every system
- Maintains error file - [1,200]ERROR.SYS
- Look at live sysdis screen
- Commands: **SYSTEM SYSdis (COMMAND) CHANGE SYSTEM DISPLAY SCREEN.**

<b>Command</b>	<b>Description</b>
<b>CLASS</b>	Class name for each job.
<b>DISK</b>	Free disk storage
<b>IDLE</b>	The time a job has been idle.
<b>JOB XX</b>	Detailed job information.
<b>MON D</b>	Disk performance of the system.
<b>NEX</b>	Advance one screen from the current display.
<b>PCT</b>	The percentage of processor being used.
<b>PER</b>	Performance statistics.
<b>PREV</b>	Previous screen.
<b>REF</b>	Clear screen.
<b>RER</b>	Clear screen and reset.
<b>SCR X</b>	Go to screen number X.



\* Characters in these fields defined on the next page



Swapping writes (avg. #/sec.)

Swapping reads (avg. #/sec.)

Item displayed in last column

IDR      NTW      PCT      0      11

34%U    18%M

16778    0.53

Virtual free core \_\_\_\_\_  
(in pages)  
Swapping Ratio \_\_\_\_\_  
Stats for PER or DISK **\*(6)**  
Time and date  
Total user core (in pages)  
Monitor version running

The following list have numbers in parentheses, which correspond with the SYSDIS screen example on the previous page.

**Baud Rate (1)**

a 110  
b 300  
c 450  
d 1200  
e 1800  
f 2400  
g 4800  
h 9600  
i 19.2K  
j 38.4K  
k 48K  
l 56K  
m 64K

**Monitor Queue (4)**

A Disk allocate wait  
B Buffered IO wait  
C Core block wait  
D Disk IO wait  
E Event wait  
F Fragmented  
H Halted (^C)  
I IO wait  
K Teletype input wait  
L Locked in core  
N Teletype input wait  
P Teletype output wait  
Q Monitor buffer wait  
R Running  
S Sleep  
U Alter UFD wait  
W Command Wait  
X Unknown

**Segment Status (3)**

# Single Segment  
\* Multiple Segment

**Core Status (5)**

F Fragmented  
L Locked  
S Swapped

**Job Status (2)**

+ High priority  
run/disk  
\* Free/marketing  
? Information  
Service User

**PERFORM Stats (6)**

I % Idle - Not in use  
L % Lost - Time lost to CPU due to bad response  
O % Overdraw - Over class max.  
U % Used - Being used by users  
M % Monitor - Used by monitor

**Network notes**

The network terminal name is in lower case unless connected via a gateway.

cty Console Teletype  
- Detached  
= Hung job  
P12j1 P = Pseudo-teletype (PTY) #12 of j1 (Job 1)

**Swapping Ratio**

**Swapping Ratio = Virtual Core in Use/Total User Core**

A swapping ratio < 1.00 means no swapping has occurred. If the swapping ratio is > 1.00, then swapping has occurred and that user core is over-committed by a percentage represented by the decimal number.

## **SYSTAT**

- Provides a typeout of the current system status
- Commands:

<b>SYS B</b>	Busy device status.
<b>SYS F (=XXX)</b>	Free disk storage status.
<b>SYS P (=XXX)</b>	Disk performance. XXX = Structure or disk drive. This will list correctable, soft, hard and rib errors.
<b>SYS R</b>	Swapping status.
<b>SYS job#</b>	Status of specific job.
<b>SYS [P,PN]</b>	Short status of specific [P,PN]

- Conditions when to use

## **SYSTEM**

- Used to control access to the system
- Used to enter SYSDIS commands on the CTY
- Used to kill jobs from the system
- Invoked by the command SYSTEM
- Subcommands:

<b>SYSTEM START</b>	START OVERHEAD JOBS.
<b>SYSTEM SHUT XXXX</b>	SHUT SYSTEM DOWN AT SPECIFIED TIME.
<b>SYSTEM SHUT STR YYY XXXX</b>	SHUT STRUCTURE YYY AT SPECIFIED TIME.
<b>SYSTEM SHUT CANcel</b>	CANCEL A PREVIOUS SYSTEM SHUTDOWN.

- Time Interval for shutdown:

<b>Time</b>	<b>Action</b>
<b>1 hour</b>	Warning message.
<b>30 minutes</b>	Warning message.
<b>15 minutes</b>	ICON answering off. Warning message.
<b>10 minutes</b>	Warning message.
<b>5 minutes</b>	DATA, REMOTE FREE answering off. Warning message.
<b>2 minutes</b>	Warning message.
<b>1 minute</b>	Warning message.
<b>0</b>	Last warning
<b>-1 minute</b>	Kills all jobs except the CTY and jobs protected by INHIBIT.

**EXAMPLES:**

```
SYSTEM SH 0500
Next time available: <cr>
Inhibit kill for: OPR,OPS
-----
System shutdown at 0500 EDT 8 Aug 1995, no "available" time specified
OPR and OPSE jobs will not be killed.
-----
1619 EDT 7 Aug 1995
OK
SYSTEM SH STR CAF 0400
Next time available: <cr>
Inhibit kill for: OPR,OPS
-----
CAF: shutdown at 0400 EDT 8 Aug 1995, no "available" time specified
OPR and OPSE jobs will not be killed.
-----
1619 EDT 7 Aug 1995
OK
```

```
SYSTEM SH CAN
No system shutdown scheduled.
-----
CAF: shutdown at 0400 EDT 8 Aug 1995, no "available" time specified
OPR and OPSE jobs will not be killed.
-----
1619 EDT 7 Aug 1995
OK
SYSTEM SH STR CAF CAN
No CAF: shutdown scheduled.
No shutdowns of any kind are scheduled.
1619 EDT 7 Aug 1995
OK
```

**SYSTEM Log xxx [ON or OFF]**

Specifies to the host which types of terminals/jobs are allowed to log in on the host. When a host allows a user to log in (answers his request to log in), "answering" is ON. Operations can restrict access to certain users or all, by using the SYSTEM LOG subcommands. If no arguments are used, the current status is given.

<b>Argument</b>	<b>Description</b>
<b>ALL</b>	All terminals/jobs
<b>DATASET</b>	Users accessing via phone lines
<b>FREE</b>	Non-billable PPNs (see REACT)
<b>ICON</b>	Independent Processing jobs
<b>LOCAL</b>	Computer room terminals only
<b>REMOTE</b>	Hardwired terminals
<b>STR xxx</b>	User access to a particular disk structure
<b>INITIAL</b>	Initial connections for that host.

### **SYSTEM Kill xxx**

Terminates the specified job(s) on a host. If multiple jobs are killed, a message is typed to verify that the operator really wants to kill the jobs. The operator types "Y" if he is sure. If the command is confirmed, the jobs are logged off by LOGOUT, which also types the message "JOB TERMINATED BY OPERATOR" on the terminals of all jobs that are killed.

This command is intended only for cases in which the specified jobs cannot be logged off via normal means (with the user's cooperation). SYSTEM KILL commands are only to be used with consent of the LEAD OPERATOR or COMPUTER OPERATIONS SUPERVISOR.

<b>Command</b>	<b>Description</b>
<b>ALL</b>	Kills all jobs. This will not kill the jobs running on the CTY, SYSDIS, OPCON, UNSPOL, or BILDAE. Such jobs can only be killed one at a time by specifying the job numbers. This command should be used only when the host is to be immediately taken down. Usually, the host's processor is stopped after the SYSTEM KILL ALL command.
<b>HUNGUP</b>	Kills all hung-up jobs. It should only be used there are no free job slots, when there are many hungup lines caused by a line failure and it is apparent that a large majority of the users are not RECOVERing.
<b>JOB#[P,PN]</b>	Kills the job specified. This command does not require confirmation since the PPN must match the job number. This command is commonly used when a user has requested via SEN OPR that another job in the same PPN is to be logged off.
<b>[PROJ,*]</b>	Kills an entire project.
<b>STR xxx</b>	Kills all jobs accessing the specified structure. "xxx" can be any structure name. This is permitted only with authorization from the Shift Supervisor.

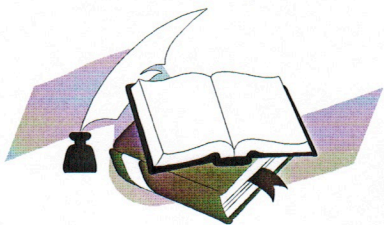
### **SYSTEM SYSdis (COMMAND)**

(See SYSDIS for commands)

### **SYSTEM STart**

Starts all system overhead jobs running from the default file [1,22]AUTSTR.CMD. STARTR reads the file and processes only the entries that match the system name, disk structures online, processor types, or center names that are in effect for the host where the program is running. If you wish to start a file that you have edited in [1,2], you must specify SYSTEM START [1,2]<file>. SYSTEM ST .SYS will use [1,22]AUTSTR.SYS, which will only start up the overhead job SYSDIS.

- Restrictions on using SYSTEM



# Review

## 1. OPSER

- Subjobs
- Importance of the ":" and "-" in commands.
- Control characters in OPSER.
- Auto-jobs

## 2. Overhead programs

- Constantly running on all systems
- Start all overhead programs using STARTR
- Automatically started when system is reloaded

## 3. Monitoring Systems

- Getting updated display of system status using SYSDIS
- Controlling access to the system using SYSTEM



## Examination Two

1. What is a subjob?
2. Give the command to OPSER that would list all subjobs logged into a CTY?
3. What overhead program constantly updates billing files?
4. What overhead program keeps track of where all structures are mounted?
5. Give the command to start all overhead jobs on a host?
6. Name the three OPSER subjobs that should always be logged into the CTY.
7. What two special characters are used in OPSER commands?
8. List three important functions that SYSDIS performs in the computer room?
9. What command is used to start the program SYSDIS running by itself?
10. What program do we use to kill jobs from a host? Give the command to do this.
11. How do you know when the next auto-job is scheduled to run?

### **SYSTEM Kill xxx**

Terminates the specified job(s) on a host. If multiple jobs are killed, a message is typed to verify that the operator really wants to kill the jobs. The operator types "Y" if he is sure. If the command is confirmed, the jobs are logged off by LOGOUT, which also types the message "JOB TERMINATED BY OPERATOR" on the terminals of all jobs that are killed.

This command is intended only for cases in which the specified jobs cannot be logged off via normal means (with the user's cooperation). SYSTEM KILL commands are only to be used with consent of the LEAD OPERATOR or COMPUTER OPERATIONS SUPERVISOR.

<b>Command</b>	<b>Description</b>
<b>ALL</b>	Kills all jobs. This will not kill the jobs running on the CTY, SYSDIS, OPCON, UNSPOL, or BILDAE. Such jobs can only be killed one at a time by specifying the job numbers. This command should be used only when the host is to be immediately taken down. Usually, the host's processor is stopped after the SYSTEM KILL ALL command.
<b>HUNGUP</b>	Kills all hung-up jobs. It should only be used there are no free job slots, when there are many hungup lines caused by a line failure and it is apparent that a large majority of the users are not RECOVERing.
<b>JOB#[P,PN]</b>	Kills the job specified. This command does not require confirmation since the PPN must match the job number. This command is commonly used when a user has requested via SEN OPR that another job in the same PPN is to be logged off.
<b>[PROJ,*]</b>	Kills an entire project.
<b>STR xxx</b>	Kills all jobs accessing the specified structure. "xxx" can be any structure name. This is permitted only with authorization from the Shift Supervisor.

### **SYSTEM SYSdis (COMMAND)**

(See SYSDIS for commands)

### **SYSTEM Start**

Starts all system overhead jobs running from the default file [1,22]AUTSTR.CMD. STARTR reads the file and processes only the entries that match the system name, disk structures online, processor types, or center names that are in effect for the host where the program is running. If you wish to start a file that you have edited in [1,2], you must specify SYSTEM START [1,2]<file>. SYSTEM ST .SYS will use [1,22]AUTSTR.SYS, which will only start up the overhead job SYSDIS.

- Restrictions on using SYSTEM

## **MAGTAPES**

### Labeling a magtape

- Different preprinted magtape labels
- Importance of magtape labeling
- When not to label a magtape

### Magtape handling

- Reel to reel drives
- VAX drives
- Cartridge drives
- Cartridge stackers
- DAT drives

### Magtape drive cleaning

## **MOUNTING AND DISMOUNTING**

### **OPCON**

- Processes magtape and disk mount and dismount requests
- How requests are queued
- OPCON commands
- Difference between magtape and disk mounts
- Importance of request number
- How to unload a disk structure
- Example:

### User Types MOUNT command

UCON makes an entry in [3,3] (the OPCON queue)

OPCON scans the queue, sends a message to the CTY:

```
*25 Mount File structure MISO: on DZA5:
      or
*26 Mount cartridge 6250BPI BISBIL[35,670]
```

Operations sees the message, looks up the cartridge (if applicable),  
mounts the cartridge or disk structure, types on the CTY:

```
OPC-25
!! Mounted MISO: for User [1,2] job 12
```

OPC-26 on MZA#: (# = the tape drive the tape was mounted on)  
 !! Magtape BISBIL[35,670] mounted on MZA10: for Job 49

USER can now access the disk or tape.  
 When USER is finished - used the DISMOUNT command.

UCON makes an entry in [3,3] the OPCON queue  
 OPCON scans the queue and types to the CTY:

\*27 Dismount magtape BISBIL [35,670]  
 (Disk structures are left mounted until Operations uses the RUL str: command)

OPC-27  
 !! Magtape BISBIL [35,670] dismounted from MZA10:

Return the cartridge to the rack.

- Other Commands:

Command	Description
<b>RUL &lt;str&gt;:</b> <b>EXAMPLE:</b> OPC-RUL CAF: OPCON processing ... ! DZA3: - Structure has been unloaded	This command will request to unload a specified structure. OPCON will not let a structure be dismounted until all users have also dismounted the structure. Make sure that OPCON says that the structure has been UNLOADED before physically removing the disk pack(s).
<b>EXIT</b>	OPCON will finish all requests that are processing and then exit to monitor mode.
<b>KILL &lt;request #&gt;;&lt;comments&gt;</b>	Cancels the specified request. The only time a request can be killed is when it is waiting on the operator to perform the task. If no comment is made then OPCON will ask for reason. This is passed on to the user as an error message. Beware of what you say!
<b>PEEK &lt;request #&gt;</b>	Examines the OPCON request queue [3,3] and types a summary of pending requests. By typing PEEK followed by a request number, the Operator can see more detailed information about the specific request. If no request number is specified, OPCON types a summary of all pending requests, along with general information about each request.
<b>STAtus x</b>	Lists the status of the functions and/or devices. The arguments are F (function) or D (devices). If no option is specified, it types out a complete status of both. (See example next page)

**EXAMPLE:**

**OPC-STA D**

OPCON processing ...

OPCON Version 1J(1465) on System CHF at 08-Aug-95 10:14:56

Device(Type)	Status	Tape	Mount Count
--------------	--------	------	-------------

DZA0: (M1800)	Active	DKF:	(47)
DZA1: (M1800)	Active	DKF:	(47)
DZA2: (M1800)	Active	NEY:	(24)
DZB0: (M1800)	Active	CAF:	(11)
DZB1: (M1800)	Active	URA:	(9)
DZB2: (M1800)	Active	NEX:	(20)
DZB3: (M1800)	Active	NEX:	(20)



# Review

1. Magtapes
  - Magtape labeling/handling
  - Magtape drive cleaning
2. Mounting and dismounting magtapes and disk drives using OPCON
  - The difference between disk and tape requests
  - OPCON commands



## Examination Three

1. What is the program that controls the mounting/dismounting of tapes and disks?
2. Explain the importance of labeling tapes/disks.
3. Explain how a magtape mount request is handled.
4. Give the command to OPCON that would unload a disk structure?
5. How can you get a status of all disks on a host?

## MAGTAPE UTILITY PROGRAMS

### FILSAV

- Provides backup for disk
- Save Commands - saves files from disk to tape.

<b>CMS</b>	Complete save. Sets a flag in each file's RIB. Done once per week.
<b>INS</b>	Incremental save. When a file is update/created, the CMS flag is cleared. The INS saves all files with the CMS flag cleared. <i>(This cuts down on cartridges)</i> Done daily, except on CMS day.
<b>SPS</b>	Special save. Saves every file specified but has nothing to do with the CMS flags. Done whenever requested, or to save temporarily to tape and restore to another host.

- Other Commands:

<b>RES</b>	Restore file(s) from cartridge to disk.
<b>DIR</b>	Creates a directory by reading the cartridge.
<b>VER</b>	Compares a FILSAV magtape with files on disk and reports any differences.
<b>@file</b>	Reads commands from "file.CMD"

- Example Commands:

INS str:[\*,\*] TO MTA?:/S (Incremental Save)  
 CMS str:[\*,\*] TO MTA?:/S (Complete Save with Date Checking)  
 SPS str:[p,pn]FILE.EXT TO MTA?:/switch (Special Save)  
 RES MTA?:[p,pn]FILE.EXT TO str:[p,pn]/switch (Restore)  
 VER MTA?:[\*,\*] WITH str:/switch (Verify)  
 DIR MTA?:[\*,\*] TO str:[p,pn]file.dir (Directory)

- Default density is 6250 BPI
- Difference between CMS and INS saves
- Difference between special and regular saves
- Switches:

Switch	Description
<b>/D</b>	Restores regardless of the disk file date. This switch tells FILSAV to overwrite any files on disk with the same name as those specified in the RESTORE command, regardless of creation date. Normally, files on disk with a creation date newer (or equal to) will NOT be overwritten by the RESTORE command.
<b>/F</b>	Prints file names to TTY (and output device)
<b>/I</b>	Ignores files that are on disk but not on the tape. FILSAV will still tell you about files that are on tape but not on disk, but not vice-versa.

Switch	Description (continued)
/L	Prints detailed directory to TTY (and output device)
/M	Marketing; changes access date to current date for Marketing PPNs [n777,*] so NITELY will not delete the files restored.
/N	No rewinds before or after completion (rarely used)
/P	Prints detailed directory to disk (str.DIR, str = structure)
/S	Suppresses directory printed to TTY so only one PPN is listed every five minutes on the output. This is helpful when doing nightly saves for CIS structures.
/U	Rewinds, but does not unload the magtape when command is completed.

**Example Run:**

```

R FILSAV
* INS DKF:[*,*] TO MTA11:/s

Save on MTA11 at 2029 09 JUL 94
INS FILSAV mag-tape number 01: 9 track, 6250 BPI
Recorded on 2029 09 JUL 94 CMS was 2048 04 JUL 94
[1,2] [1,4] [1,6] [3,6]
[340,341]
%FSVPET Physical EOT at 2037 09 JUL 94
** CON MTA10:
[340,400] [340,500] [340,6000]
[677,1000]
SAVE COMPLETED
*EXIT

```

- Sub commands

When FILSAV encounters the end of tape or an error, a message will type out, and the operator will be given a double asterisk (\*\*) prompt. The "\*\*\*\*" prompt indicates that FILSAV is waiting for one on the following commands to tell it how to recover from the situation.

Sub-command	Description
KILL	Stop immediately and return to FILSAV'S main command level.
HALT	During a read, the requested directory file is closed. During a write, it backspaces to the beginning of the current file, writes the trailer record, and returns to FILSAV'S main command level.
SKIP n	During read, it skips "n" files. This may be used to skip over a bad file.
RETry	Retry the read/write function if possible. If a save gets a PARITY ERROR at the beginning of the tape, a new tape can replace it and the RET command will except it. Do not RETRY on a CMS save, restart it.
CON MTA?:	Continue to read from/write to the next tape. Used to specify a continuation to another tape drive. Also used to ignore read/write error and continue.
IGNore	Ignore error and continue if possible

- **ERRORS**

<b>Error</b>	<b>Description</b>
<b>CHECKSUM ERROR</b>	If you attempt a restore and the program encounters a CHECKSUM error, FILSAV will inform you of the error and ask if you want to continue with the restore.  If you answer "yes", you will receive a message saying that you have probably just restored a bad file.
<b>DATA ERROR</b>	A data error could be a hardware problem with the tape drive or controller, or a problem with the magtape.  Try a different magtape and clean the tape drive. If problem persists, fill out a STC Trouble Report, and if necessary, contact ISG.
<b>PARITY ERROR ON MTA?:</b>	A parity error on the tape drive usually signifies either a bad magtape or a dirty tape drive.  Try a different tape and/or clean the magtape drive. Don't forget to place a label on the tape if it needs to be cleaned or destroyed.

- **Tape rotations:**

<b>DAILY</b>	Kept 7 days.
<b>WEEKLY</b>	Kept 8 weeks.
<b>MONTHLY</b>	Kept 1 year.
<b>BOND</b>	Kept 45 days.

- **FILSAV restores**

### **FILSAV SCHEDULE (See the daily Save Schedule)**

<b>Arlington</b>	
22:00	I?? (With at least 25% idle time.)
12:00	Primary structures. ( Except European Hosts )
17:00	Most permanent structures except:
18:00	BNK, URA
20:00	NEX, ZZZ, European Host structures
<b>NOTE:</b> Make a duplicate save of BPM and send it to Dublin; BIL and DAY, after billing; SSA, after SOP240. Do NOT log into TAPES, send to Dublin.	
<b>Dublin</b>	
19:30	I?? (With at least 25% idle time.)
12:00	FD?, SD?, primary structures
17:00	Most other structures except:
18:00	DAJ
22:00	VSA, European Host structures
<b>NOTE:</b> SSRA after SSR sources are gathered (check with SSR Admin.). Do NOT log into TAPES, send to Arlington.	

## **BSDI Saves**

See *BSDi Save Book*

## **Windows NT Saves**

### **ARCADA**

#### **INTRODUCTION**

Arcada is used to backup the computer room's NT servers. This document contains the procedures to use this program.

#### **DAILY SAVES**

Last Update: 1/10/97

File Name: \PROCEDUR\32BIT\ARCDAILY.DOC

#### **INTRODUCTION**

This procedure shows the steps to schedule a *normal* daily save for NT servers being backed up.

#### **SCHEDULING A SAVE**

1. Log into the correct Arcada Backup Server for the machine that is being backed up.  
**NOTE:** To find the correct backup server to use, see G:\opr\backups\arcadabu.xls.
2. Start up Backup Exec.
3. Click on **J**obs which is located on the title bar and select **S**etup. This will bring up the Job-Setup window.  
**NOTE:** The dat tapes that are being used for a save must be erased before a save can be done. This involves scheduling an erase job, then scheduling the save. Be very careful not to schedule the save to begin before the erase job.
4. Highlight the job you want to run (i.e. ERASE-T1, LOTUS1-INS-T1, LOTUS1-CMS-T1, etc...) and click on the **S**chedule button. At this time, the *Schedule a Job* window pops up.
5. Select **A**dd to schedule a save job.
6. From the *Schedule Options* window, change the Repeat Interval to **Only Once** by clicking on the radio button next to that option.
7. Click **OK**.
8. Repeat steps 4 - 7 to schedule more jobs.  
**NOTE:** Remember to schedule an erase job to start first, then the save.
9. Once the job(s) have been scheduled, you must exit out of Backup Exec for the jobs to begin.

**NOTE:** To check on the status of Backup Exec jobs, bring up the Backup Exec View.

10. Once the job(s) are finished, label the tape cases with the filsav label. Be sure the permanent number on the dat tape is written on the filsav label in case the dat tape falls out of the tape case.

END OF PROCEDURE

### **SPECIAL SAVE**

Last Update: 1/17/96

File Name: \PROCEDURE\32BIT\ARCADSAV.DOC

### **INTRODUCTION**

This is the process to create a special save using ARCADA in the Windows NT environment.

### **STEPS TO CREATE A SPECIAL SAVE**

1. Log into backup account on a BACKUP EXEC machine.
2. Erase the tape(s) you're planning on using during the save.
3. Select the **"backup selections"** icon from the bottom of the screen.
4. Select the servers, drives and or files you wish to save.
5. Select the **BACKUP** "button" from near the top of the screen.
6. You'll need to provide the following information:
  - a) Job name
  - b) Select **overwrite the tape**.
  - c) Give a tape name and label.
  - d) Under backup options, method=Copy (for an SPS)
7. Select **Advanced Options**
  - a) Verify tape after backup
  - b) compression should be **hardware**
  - c) Skip open file = NO
  - d) Select OK
8. Select the **RUN NOW** button, this will run the backup in the foreground so you can watch it to see when it's done.

END OF PROCEDURE

## **RESTORE**

Last Update: 3/17/97

File Name: \PROCEDURE\32BIT\ARCADRES.DOC

### **INTRODUCTION**

This is the process to restore from a special save using ARCADEA in the Windows NT environment.

### **STEPS TO RESTORE FROM ARCADEA SAVES**

1. Log into backup account on a BACKUP EXEC machine.
2. Put the tape that has files to be restored in a tape stacker on this machine.  
(Don't press the tape load button on the drive, Arcada will load the tape)
3. Select **Operations** from the top menu bar.
  - a) Choose **Catalog Media**.
  - b) Change the **Device** to the correct tape drive.
  - c) Select **Run Now** to catalog the tape.
  - d) Select **OK** when the catalog is complete.
4. Select the actual tape you are using from the left side of the current window. **Double click on the words** beside this icon (**CMS PROD** etc.)

**NOTE:** DO NOT make a check mark in the box, this will tell Arcada that you want to restore the entire tape. The tape indicator on the left part of the screen may indicate it's tape 1, even if you have the tape to be restored in drive 2. This is simply the drive it was saved on (it's part of the tape label).

- a) From the right side of the window, select the server name you want to restore from.  
(Double click on the directories to tree down.)
  - b) Put check marks in the boxes beside the files that are to be restored.
5. Click on **Restore** from the toolbar under the top menu bar.
  - a) Change the **Job Name** to something that is easy to remember (i.e. PROD1 RESTORE).
  - b) If you are restoring file(s) back to their original location, leave **Type of Sets to Redirect** at a value of **None**. Otherwise, click the radio button next to **File Sets**.
  - c) If **File Sets** was selected in the **File Set Destination** area:
    - i) Type in the **server** and **share** you want to restore to.
    - ii) Type in the **Path name** to restore to in the **Restore to Path box**.
  - d) If the **Registry** is to be restored, check the **Restore Registry** box.
  - e) In the hardware section, verify the **Device** is the correct tape drive.
  - f) Select the **Advanced** button.
  - g) Choose an option under **Restoring Over Existing Files**.

**NOTE:** This depends on the request that is being done.
  - h) Select **OK**.
6. Click **Run Now** to start the restore.
7. Watch the **Job Status** window, when it comes back with "Operation Successfully Completed", click **OK**.

8. Exit out of **Backup Exec**.
9. Check to make sure the files were restored to the machine specified.

END OF PROCEDURE

## **RESTARTING AN ARCADA BACKUP**

Created: 1/22/97 JS    Last Update: 3/14/97 DWN    File Name: \PROCEDUR\32BIT\ARCRESTA.DOC
--

### **INTRODUCTION**

This procedure is to be used when an Arcada backup bombs during the process of a save.

### **RESTARTING THE SAVE**

1. Check the log file to find out which *server* was being backed up when the save died. You will also need to note which *share* it was on.
2. On the LAN, open the ARCADABU.XLS file located on G:\OPR\BACKUPS. Under Edit on the drop down menu, do a search for the server name. Starting with the server\share the backup died on, list all the remaining servers and shares that are to be saved on that particular backup.
3. From the Arcada server where the backup is to be run, open the Backup Exec window and create a new selection list using the following steps:
  - Double click the **Backup Selections** icon.
  - Double click **Networks** (the word, not the box).
  - Double click **Microsoft Windows Network** (the word, not the box).
  - Depending on the machines that need to be backed up, double click the appropriate group, (PROD, WWW, etc....again, the word, not the box).
  - Double click the machine names that need to be backed up (once again, not the box!).
  - Click on the box beside the shares that are to be backed up.
  - Double click the machine name again to close it and continue with the next machine name until all machines for that backup have been selected.
  - After all servers to be backed up have been selected, choose **Select** from the drop down menu.
  - Next, click on **Save Selections**...you will be prompted for a name (you may name your selection list anything as long as it's not already being used). If you choose a name that's already in use, it will ask if you want to overwrite the existing selection list.

<b>NOTE: BE CAREFUL NOT TO OVERWRITE THE ORIGINAL SELECTION LIST!</b>
---

4. After saving your selection list, you will need to create a job for the backup. From the drop down menu, choose **Jobs**, then **Setup**. Under **Operations**, choose **Create**.

<b>Job Name:</b>	Use something descriptive, such as PROD1-CMS-RESTART-T1.
<b>Operation:</b>	BACKUP (default)
<b>Selection List:</b>	Whatever you named your selection list in the above steps.

- Click **Options**, make sure **Overwrite the media** is checked

<b>Label:</b>	Something descriptive, can be the same as the job name above.
---------------	---

- Under **Backup Methods:**

<b>Files:</b>	NORMAL (for CMS) or DIFFERENTIAL (for INS)
---------------	--

- Under **Hardware**:

<b>Device:</b>	T1 or T2	
<b>Magazine:</b>	<b>Standard :</b>	For a 6 tape stacker (This will also change Group).
	<b>Archive:</b>	For a 4 tape stacker.

- Click **Advanced...** Under **Compression**, select **Hardware**

5. Schedule the job you just created.

END OF PROCEDURE

### **MAGTAP**

- Used to position magtapes
- MAG REW MTA?
- MAG UNL MTA?

### **MAGDUP**

- Used to duplicate data on a magtape
- Dialogue:

```
R SPT:MAGDUP
```

```
Buffer size (words): 10000
```

```
Input drive: MTA10:
```

```
Input density: 1600
```

```
Output drive: MTA12:
```

```
Output density: 6250
```

```
Rewind the tapes (Y or N): Y
```

```
Stop at logical EOT (Y or N): Y
```

```
Rewind unload after done (Y or N): Y
```

```
Industry compatible mode (Y or N): N
```

- FILSAV and BACKUP are NOT industry compatible
- @DUPTAP

## TAPES

- Runs from [100,4]
- Maintains Magtape library and FILSAV library
- Tape rotations:

MAGLIB		FILSAV	
<b>T</b>	TEMP. - KEEP AT MOST 1 MONTH	<b>DAILY</b>	Kept 7 days.
<b>R</b>	REGULAR - KEPT UNTIL EXP. DATE	<b>WEEKLY</b>	Kept 8 weeks.
<b>P</b>	PERMANENT - KEPT FOREVER	<b>MONTHLY</b>	Kept 1 year.
<b>C</b>	CARTRIDGE (Cartridge and DAT tapes)	<b>BOND</b>	Kept 45 days.
<b>K</b>	CANISTER (Tapes in canisters - rare)		
<b>H</b>	HANGING (reel to reel)		

- Commands for inserting, finding, and releasing:

<b>INS M</b>	INSERT A TAPE INTO MAGTAPE LIBRARY (WILL PROMPT)
--------------	--

**EXAMPLE:**

```

r spt:tapes
Center: arl
Center: COLUMBUS
ARL+ ins m
PPN: 140,26
Kanister, Hanging, or Cartridge? C
Regular, Permanent or Temp? R
Tape ID: TRAIN
Description: SPS of CAF:[140,26]
Date: 8/ 8/1995 :? <cr>
Volume: 1
Owner: O (o = ours, t = theirs)
Expiration date: 1/1/96
Expiration instructions: Release and scratch
Structure: CAF

Processing...
WARNING: NO empty permanent canister slots in MAGLIB
WARNING: NO empty regular canister slots in MAGLIB
WARNING: NO empty temporary hanging slots in MAGLIB
Completed
Tape entered in slot #21398
  
```

<b>INS F</b>	INSERT FILSAV TAPES (WILL PROMPT FOR STR AND PPNS)
--------------	--

**EXAMPLE:**

```

ARL+ ins f
Date: 8/ 7/1995 ? <cr>
Structure: CAF
This is a Daily, INS filsav
Number of volumes: 2
STR:CAF, Type:Daily INS, Vol:1 - 2
Enter in slots: 50241 - 50242
Structure: -<cr>
ARL+
  
```

FIN M[P,PN]'TAPEID'	***WARNING*** DO NOT ENTER <b>FIND M</b> <CR>, TYPE IN THE ENTIRE COMMAND LINE OR YOU WILL GET A LISTING OF THE ENTIRE LIBRARY. (:XP MAY STOP THE TYPEOUT).
FIN F	FIND SLOT NUMBER OF A FILSAV TAPE (WILL PROMPT)
ALL F/STR:XXX	FIND ALL FILSAV TAPES FOR THAT STRUCTURE.
ALL F/DA:XX/XX	FIND ALL FILSAV TAPES FOR THAT DATE.
REL M	RELEASE A TAPE FROM MAG. LIBRARY (WILL PROMPT)
REL M#xxxxxx	RELEASE TAPE FROM SPECIFIC SLOT NUMBER (FASTER!) <b>EXAMPLE:</b> <pre> ARL+ rel m#21398  #21398  [140,26] 'TRAIN '   Vol:1   Aug-08-1995 Owner:O Type:R Media:C STR:C &lt;Aug-08-1995 %Aug-08-1995 @Jan-01-96 (Release and scratch) "SPS of CAF: [140,26] "  This record? y  Processing... Completed Slot #21398 released (Release and scratch) Is this tape sold(Y,N,G,X)? h  Y = Compuserve tapes that are sent out and sold. N = Customer tapes that are sent out. G = Compuserve tapes that are sent out, but not sold. X = Compuserve or customer tapes that are not sent out.  Is this tape sold(Y,N,G,X)? x Tape is not sent. </pre>
PULL	RELEASE EXPIRED FILSAV TAPES, DONE AT 12:01 DAILY.
SEND	KEEPS TRACK OF TAPES SENT OUT OF COMPUSERVE.
MOU M#xxxxxx	TO UPDATE THE LAST DATE THE TAPE WAS MOUNTED.

## TASKS

- Maintains billing for tasks performed.
- Operators report scheduled and form tasks
- Runs from [100,4]
- Commands:

Sxxxx	TASK OFF SCHEDULED TASK # xxxx.
Fxxxxx	TASK OFF FORM # xxxxx.
CHA Fxxxxx	ALLOWS CHANGES TO BE MADE TO A TASK COMPLETED WITHIN THE LAST 6 HOURS. SCHEDULED TASKS MAY ALSO BE CHANGED.
CHE Fxxxxx	LISTS COMPLETED TASK (OR RANGE OF TASKS).
L [P,PN]	LISTS COMPLETED TASKS THAT MATCH THE SPECIFIED PPN.
P n	LISTS PENDING TASKS FOR THE NEXT 'n' HOURS.

(See Examples on the next page.)

**Examples:**

```
R TASKS
Center: ARL

Operation`s Task System - Center: Columbus

09-Feb-96 10:44
+ S7530
Name: DKH
OK ? Y
TTY: -
Enter remarks (max. 20 lines)

[ Entering editor - type /HELP for help, /EX to exit ]

[ Top of file ]
OK
/EX

Flag ? N

09-Feb-96 10:45
+ F153272
Name: DWN
OK ? Y
Chargeable: N
PPN: 140,1
D1: Type of task (SPS, RES, DIR, SOPnnn, TTD, DTT, OPR): OPR
D2: Number of tapes sold (if any): 0
Enter TASK description (max. 8 lines)

[ Entering editor - type /HELP for help, /EX to exit ]

[ Top of file ]
THIS WAS A TASK TO SHOW AN EXAMPLE OF WHAT A FORMED TASK
LOOKS LIKE
/EX

TTY: -
Enter remarks (max. 20 lines)

[ Entering editor - type /HELP for help, /EX to exit ]

[ Top of file ]
OK
/EX

Flag ? N

09-Feb-96 10:46
+ EXIT
```

## **BACKUP**

- DEC magtape utility program
- Advantage of "save sets"
- Uses in Operations
- Commands:

<b>TAPE MTA?</b>	TAPE DRIVE SPECIFIED FOR THE SAVE.
<b>FILES</b>	PRINT THE NAME OF THE FILE(S) DURING THE RUN OF BACKUP.
<b>DEN 6250</b>	SET THE DENSITY TO 6250.
<b>SSN XXXXXX</b>	SAVE SET NAME ("ALL" IF RESTORING ENTIRE TAPE)
<b>EOT</b>	SKIPS TO END OF TAPE (FOR APPEND)
<b>SKIP (-)n</b>	SKIPS n FILES FORWARD OR BACKWARD (+ OR -)
<b>REWInd</b>	REWIND TO BEGINNING OF TAPE

**SAV STR:[P,PN]FILE1.EXT,STR:[P,PN]FILE2.EXT**

**Example:**

```
R BACKUP
/TAPE MTA10:
/SAV DKF: [141,2] FORMS.*,DKF: [141,2] *.DAT
!141,2          DKF
"done
OK
```

**RES STR:[P,PN]OUTPUT.EXT = STR:[P,PN]INPUT.EXT**

**Example:**

```
R BACKUP
/TAPE MTA10:
/SSN APRIL
/SUP ALW
/RES CAM: [140,1] *.*=*.DAT
" done
OK
```

**NPRI STR:[P,PN]FILE.DIR (CREATE A DIRECTORY FILE OF WHAT'S ON TAPE)**

**Example:**

```
R BACKUP
/TAPE MTA10:
/NPRI STR: [PPN] FILE.EXT
" done
OK
```

**CHEck STR:[P,PN]\*.\* (VERFIY THAT TAPE IS SAME AS DISK)**

**Example:**

```
R BACKUP
/TAPE MTA10:
/CHE DKH: [27200,12] *.*
" done
OK
```

- **Interchange mode**

BACKUP is designed to run in one of two modes. Interchange mode is used for data transfer between different sites. In this mode BACKUP does not write the specific system information on the tape, only the information critical to the file itself. On a restore, system defaults are applied.



## Review

### 1. Magtape usage programs

- FILSAV
- BACKUP
- ARCADA
- MAGTAP
- MAGDUP

### 2. TAPES

- Maintains magtape library
- Runs out of [100,4]
- Records FILSAV, MAGLIB, BILLING magtapes

### 3. TASKS

- Scheduled and form
- Runs out of [100,4]



## Examination Four

1. List the two types of tasks we must report to the program TASKS?
2. Explain why the programs TAPES and TASKS are located on one system only, rather than every system?
3. Please list the command to TAPES that would log a daily FILSAV tape into the TAPES database?
4. List 6 different FILSAV commands?
5. What does the /D in FILSAV refer to?
6. Explain the most important difference between BACKUP and FILSAV?

## **COMMAND FILES**

### **.CMD files**

- Run from only certain program prompts by typing @<file>
- Consists of a list of commands used by the program running

### **.COM files**

A COM file is a file that contains monitor (ICS) commands and/or input to programs. COM files are used to automate tasks that would otherwise be performed manually.

- In [1,103]
- Run from the OK prompt by typing @<file>
- Like a program
- Special command syntax:
  - All ICS (Monitor) commands begin with a period (.)
  - All input commands to a program begin with an asterisk (\*) or an equal sign (=). The difference is that the equal sign will suppress the carriage return.
  - Comment lines may begin with either a semicolon (;) or an exclamation point (!). Comments using (!) will type out when running the COM file.
  - Control characters begin with a caret (^).

#### **Example:**

```
! This is a comment
; This is also a comment!
.R FILSAV
*SPS CAF:[140,26]TRAIN.* to MZA12:/L/U
*VER MZA12:[*,*] WITH CAF:/I
*EXIT
.DEL CAF:[140,26]TRAIN.*
.R CONMAG
=MTA10:/MOD:EBC/PRE:400/BLO:10 TO
[131,1]OPR.DAT^[
; The ^[ is an ESCAPE
*EXIT
^C
```

### **.ATO files**

- In [103,700]
- Run by OPSEER every 2 hours
- Run manually by typing :A ATxxxx.ATO[103,700] (xxxx = time the ATO job was supposed to run)

## **SYSTEM SOFTWARE REQUEST**

- What is an SSR
- Processing procedures for automatic SSRs
- Five functions of an SSR:
  - DELETE
  - COPY
  - PROPAGATE
  - FILSAV RESTOR
  - LEVEL CHANGE
- Examples of various SSRs:
- Outline of Do-Immediate SSR processing:

### **GUIDELINES FOR PROCESSING A DO-IMMEDIATE SSR**

- The original SSR is usually hand carried to CCR, and a copy of the original SSR is sent via InfoPlex from SSRPRO to DCR.
- Any SSR forwarded from SSRPRO to DCR via IPX could have additional information concerning processing listed.
- Always check "DO" box for when to process the SSR
- Check for any special instructions on the form
- Do a DIR/A/B/L for the file(s) and compare with the directory on the SSR form before copying:
  - If the actual blocks from the DIRectory disagrees with what is listed on the form, note the actual blocks, initial the form, and continue processing.
  - If creation date/time disagrees with what is listed on the form, contact Software Administration before processing.
  - Make sure the protection level is correct before the file(s) are copied to the system area. This can have an impact on the system! (Example: If LOGOUT.EXE has the wrong protection level, users may not be able to log off!)
- Always make sure that you COPY/RESTORE/DELETE the correct files to the correct system areas and structures. Example: A file copied to [1,4] but on the wrong structure, may have over-written another version of that file! So you would have to restore the old version, not just delete the file!
- Check off each structure as it is processed so none are forgotten or processed where they should not have been.
- Check the "Proc" box at the bottom of the form and initial and date/time.

## **DEFINITIONS**

- <ssr no.> - The SSR number listed in the top right-hand corner of the form. Do not use the number assigned by the auto-SSR system.
- (pro) - The protection level to be for the file(s) as listed on the form. DEC protection level may be specified as <pro>.
- <str> or <str(s)>: - Refers to a structure, or structures to be processed.
- [p,pn] - Refers to a Project Programmer Number. For Level Change, the PPNs are represented as [1,4a] and [1,4b].
- <file(s)> - Represents the file or files to be processed.
- <pri>: - Is the primary structure on a host with structure(s) to be processed.

## **DO-IMMEDIATE COPY**

Copy from a source .PAK file to a system area.

1. Must create a backup of the source .PAK file and PAC RET the actual system files from the backup. Generally, CCR does this, and then calls DCR to inform them of what host and PPN the file is on. Use the following .COM file:

**@SSRBAK** (and answer the questions)

2. Note the <str>:[104,3]?<ssr no.>.PAK on the form.
3. When a Source Pack file is to be deleted, place a [ ] beside it and DO NOT DELETE the file.
4. **DIR/A/B/L <file(s)>** and compare with the directory on the SSR form. (should be the same)
6. Copy the file(s) to the designated system area on the specified structures. CCR then processes structures in Arlington, and DCR processes Dublin structures.

**COPY [131,1]<file(s)> TO <str(s)>:[p,pn]\*.\*(pro)**

Or for remote copy:

**COPY [131,1]<file(s)> TO <pri>:[131,1]\*.\*(pro)/REMOTE**

**COPY <pri>:[131,1]<file(s)> TO <str(s)>:[p,pn]\*.\*(pro)**

**NOTE** - Group files in the COPY by protection level.

**NOTE** - After the file(s) are in the destination PPN on one structure, FILSAV can be used to copy the file(s) to the PPN on the other structures.

## **R FILSAV**

**SPS <str>:[p,pn]<file(s)> TO MTA?:/U**

On structures to be processed:

**@RESSTR**

## **DO-IMMEDIATE LEVEL CHANGE**

A level change is basically the copying of a file from one software level to another.

Level 0 [1,4] LV0:, SYS:

Level 1 [1,41] LV1:

Level 2 [1,42] LV2:

Level 3 [1,43] LV3:

Level 4 [1,44] LV4:

Level 5 [1,45] LV5:

Level -1 [1,46] LV6:

1. **COPY <str>:[1,4a]<file(s)> TO <str(s)>:[1,4b]\*.\*(pro)**

Or for remote copy:

**COPY <str>:[1,4a]<file(s)> TO <pri>:[131,1]\*.\*(pro)/R**

**COPY <pri>:[131,1]<file(s)> TO <str(s)>:[1,4b]\*.\*(pro)**

**NOTE** - Use the file from the source structure to copy to the other structures. Do NOT just COPY from [1,4a] on each host since file versions may differ.

**NOTE** - After the file(s) are in the destination [1,4b] on one structure, FILSAV can be used to copy the file(s) to the [1,4b] on the other structures.

**R FILSAV**

**SPS <str>:[1,4b]<file(s)> TO MTA?:/U/L**

On structures to process:

**@RESSTR**

2. If DELETE AFTER TRANSFER is specified, the file(s) must be deleted from PPN (A) on ALL structures to be processed:

**DEL <str>:[1,4a]<file(s)>**

## **DO-IMMEDIATE DELETE**

Delete the file(s) from the specified structures.

1. On structure to be processed:

**DEL <str>:[p,pn]<file(s)>**

### **DO-IMMEDIATE FILSAV RESTORE**

1. Look up the magtape from the specified date.
2. RESTORE the file(s) /D to all specified structures.

#### **R FILSAV**

**RES MTA?:[p,pn]<file(s)> TO <pri>:[131,1]/D/L**

**EXIT**

**COPY <pri>:[131,1]<file(s)> TO <str(s)>:[p,pn]\*.\*(pro)**

**NOTE** - Use the tape for the structure and date specified on the form. The PPN for the system area may be different from the PPN to restore the file from.

**NOTE** - Group files in the COPY by protection level.

**NOTE** - After the file(s) are in the destination PPN on one structure, FILSAV can be used to copy file(s) to the PPN on the other structures.

#### **R FILSAV**

**SPS <str>:[p,pn]<file(s)> TO MTA?:/U**

On structures to be processed:

**RES MTA?:[p,pn] TO <str>:[p,pn]/D/U/L**

### **DO-IMMEDIATE PROPAGATE**

The file(s) are already in a specified PPN on one structure, and need to be copied to the same PPN on the specified structures. The source and system PPNs should be the same.

1. **COPY <file(s)> TO <str(s)>:[p,pn]\*.\*(pro)**

Or for remote copy:

**COPY <file(s)> TO <pri>:[131,1]\*.\*(pro)/REMOTE**

**COPY <pri>:[131,1]<file(s)> TO <str(s)>:[p,pn]\*.\*(pro)**

**NOTE** - Both P,PNs for a propagate should be the same.

**NOTE** - Group files in the COPY by protection level.

**NOTE** - After the file(s) are in the destination on one structure, FILSAV can be used to copy file(s) to the PPN on the other structures.

#### **R FILSAV**

**SPS <str>:[p,pn]<file(s)> TO MTA?:/U**

On structures to be processed:

**@RESSTR**

- **Outline of 32 bit SSR processing:**

SEE SSSSSR.DOC in the Procedure Manual under DAILY Procedures.

- **Importance of SSR successful completion**

## **INDEPENDENT PROCESSING**

Independent Processing allows users to execute jobs without interacting with the system during the job's run time. This is useful, since a user can work during the day and still take advantage of the lower night rates by requesting a night processing priority on his independent jobs. No more than two jobs for one project will run at a time. Independent jobs are entered into a queue file JOBFIL.ABT in area [3,7], which can be examined by INDQ. Independent jobs may not be run from [1,2].

To utilize Independent Processing, the user creates a file that contains the desired system commands (the extension is **.ICD** for Independent Command Dialogue). When processing starts, the specified command file is initialized. During execution a response file is created containing all the dialogue that occurred between the command file and system. This file has the same file name, but with an extension of **.IRF**, for Independent Response File. This file can be printed and examined.

Independent jobs can be set up to repeat at specified times. The System Administrator does this by making an entry in a file RPTJOB.ABT[3,7]. The format of the file, and the information fields that must be specified are described in detail in the documentation for the program RPTJOB[1,22], which is used to verify that the file is syntactically correct.

- Controls independent jobs
- Synonyms: ICDs, ICON jobs, batch jobs

## **IQUER**

- Different file extensions
  - .ICD
  - .IPR
  - .IRF
- Invoked by command IND <command>
- Possible arguments ENT, KIL, DEL, STA:

### **IND DElete <file>**

Jobs entered in the queue but not yet running can be deleted. If the job is running, use the KILL command.

#### **Example:**

```
ind del dennis
DENNIS deleted
OK
```

### **IND Enter <file>/<switches>**

This command is used to start the specified independent job. The following switches are used for special priorities or start times.

<b>/A</b>	Normal time-sharing priority. (default)
<b>/B</b>	Background priority; jobs start 2 hours after they have been entered - this is even if the start time has been specified in the ENTER command.
<b>/C</b>	Non-prime time priority.
<b>/N</b>	Notify the user's TTY upon completion of the job.
<b>/hhmmS</b>	Indicate start time (hhmm is the format for hours and minutes, i.e. 1930S) or /time:hh:mm.
<b>/WAITFOR:&lt;file&gt;</b>	Will wait for another job to complete before running the job just entered.

#### **Example:**

```
ind ent dennis/time:09:10
DENNIS entered with start time 09:10 EST 15-Mar-96

OK
ind ent dennis
DENNIS entered with start time 09:02 EST 15-Mar-96

OK
```

### **IND KILL <file>**

Jobs can be deleted from the queue by an IND KILL command. The job can be killed at anytime during the processing by using the KILL command.

#### **Example:**

```
ind kill dennis
DENNIS killed [job was running]

OK
ind kill dennis
Output now in DENNIS.IRF

OK
```

### **IND STAtus <file>**

This will check the status of an independent job. If the file name is not specified, a status of all jobs for the PPN will be listed.

#### **Example:**

```
ind sta
DENNIS waiting for specified start time at 09:10 EST 15-Mar-96
OK
ind sta
% IQRNQE - No independent job entries.

OK
```

- Independent processing queue [3,7]

## **ICON**

- Processes independent jobs that are queued by users
- Automatic processing [3,7]RPTJOB.ABT
- Normal processing [3,7]JOBFIL.ABT
- Overhead job that runs on all systems

## **INDQ**

- Used to check the status of the independent queue
- Can use to check complete system or specified projects
- Run from [1,2]
- Use IND STA command if logged into PPN that issued the ICON job
- Example:

```
R SPT:INDQ
Structure (<cr> for all) ? CBG
Project (<cr> for all) ? 310

Dump of CBG:[3,7]JOBFIL.ABT:
  PPN      Account ID  Job Name Job#  P  STS  Start
  310,2220  JLC        *DEC02   21  A  RUN  15:56 02-Dec-88
class = .INH last checkpoint: START
  310,11    JLC        X              C  WT   18:00 13-Dec-88
class = .INH
OK
```

## **SPOOLQ**

- Examines printer queue
- Subcommands:

When a subcommand lists more than one location, the output is first sorted so that it is easier to read.

**DELeTe** Allows deletion of specified spooled files. The command is not valid for [1,2].

**TYPe** Type up to 400 characters from specified spooled file.

### **The following subcommands are valid for [1,2] only:**

**LCO** Displays COL, FCS, CLR, BIG, and INH listings.

**OPR** Displays COL, FCS, BIG, DUB, and COM listings.

**COL** Files in Columbus queue only.

### **Switches:**

**/LOC:???** Lists all files for specified location, provided that location is a valid line printer. Valid for [1,2] only.

/CITY:??? Works the same as /LOC:??? above.

- **Example:**

SPO/LOC:COM			
System ID	Chars/Cpys	Spooled at	Form #
COMHZT.LPT	1280/2	11:34 EDT 9/12/97	18
Total Chars 1280			
OK			

## COMSPL

- Used to transfer microfiche files from the printer queue to magtape
- Difference between indexed and non-indexed
- When finished, renames files to \*.CBK
- Options:

<b>OVERwrite</b>	START TO SAVE (OVERWRITE) MICROFICHE TO TAPE.
<b>APPend</b>	APPENDS FILES TO THE END OF EXISTING TAPE.

- **Example:**

```
R SPT:COMSPL

COMSPL V 2(163) 30-Jan-95 08:44

Output Tape:  MTA10:
Density (800 ,1600 or 6250): 6250
Options:  OVE
Process indexed files, (form 44)? N

Label this tape as:  T30FLC
Spooled Files:

COM File:  DKE:COMKSM.LPT   Originating PPN: [1,2]           1 Fiche
File Successfully Copied
COM File:  DKE:COM80C.LPT   Originating PPN: [1,2]           3 Fiche
File Successfully Copied

*****
4 files copied to tape T30FLC
*****

[ CMLOFW - Indexed files waiting]
The COM Service Order Form and this tape should be labeled Non-Indexed.
Comspl complete 30-Jan-95 08:45
OK
```



# Review

## 1. Command files

- .CMD files
- .COM files
- .ATO files

## 2. SSRs

- Nightly processing
- Do-immediate SSRs

## 3. Independent processing

- Files with extensions of .ICD
- Four commands using IQUER
- Independent processing queue
- Program that runs all .ICDs is ICON
- Checking the ICON queue using INDQ

## 4. Printer files

- Printer queue [3,6]
- Examining the printer queue with SPOOLQ
- Different locations of files

## 5. COMSPL



## Examination Five

1. What PPN is referred to as the printer queue?
2. Give the command to check for all microfiche files located on a system?
3. What program do you use to remove microfiche files from the printer queue and save them to magtape?
4. The command MAG REW MTA10: runs what program?
5. What does MAGDUP do?
6. What are the 5 functions of an SSR?
7. What file extension will ICON give to the log file of an ICON job that has completed?
8. What program do we use to check the ICON queue?

## **BILLING PROGRAMS**

- Billing procedure is **SOP264**

### **BLOCKS**

- Builds files for each structure: as DSK:[1,200]str.BLK
- Disk storage report for billing
- Run from 16:00 ATO job

### **BILDAY**

- Collect each system's billing information for the day
- Uses system billing files in [1,200]
- When finished, runs @BILXFR.COM[1,177777]
  - Runs ISDDAY (creates a CIS billing file overhead job ISDTRK)
- When finished, runs @BLXFR2.COM[1,177777]
- Creates a single file (BIL?H?.PAK)
- Error situations

### **NTXDAY**

- Collects all Network billing information for the day
- Creates NET?H?.PAK file
- Error situations

- **Example of BILING.COM**

```
18:00:49 @BILING
18:00:49 Blday %5A(426)-2, Monday, August 7, 1995 6:00:49 PM EDT (CHF=DKF)
18:00:49
18:00:49 Operator ID: Normal
18:00:50
18:00:50 Normal daily run? Yes
18:00:50
18:00:50 Deleting BILTMP.*[1,200]
18:00:50
18:00:50 Catalogue [1,200]
18:00:50 TAPES.DEV LOANUM.SYS MOUNT.DEV HOLIDAY.DAT SYSINF.BKP
18:00:51 OPCON.DEV COMLBL.SYS TERTYP.DAE ISDMIS.DAE NNAMES.BIN
18:00:51 FTPD.LOG WCMCHF.D08 WCMCHF.D09 WCMCHF.D10 WCMCHF.D11
18:00:51 WHOCIM.LOG WCMCHF.D23 WCMCHF.D24 WCMCHF.E1 WCMCHF.E11
18:00:52 BLYBOB.DAE WCMCHF.F21 WCMCHF.G31 WCMCHF.H01 WCMCHF.H02
18:00:52 WCMCHF.H03 WCMCHF.H04 WCMCHF.H05 OPCON.LOG WCMCHF.H06
18:00:52 FACTX.DAE HAYDN.DAE CHARGX.DAE DEVICX.DAE MISSLE.DAE
18:00:52 BADLOG.DAE YXFER.SYS WHOCIM.DAX RESPON.DAT DKF.BLK
```

```
18:00:52 NEY.BLK    CAF.BLK    URA.BLK    NEX.BLK    ERROR.SYS
18:00:52
18:00:54
18:00:54 $ Past Normal daily Run !
18:01:10
18:01:10 ! Storage date: 07-Aug-95
18:01:12 8/7/1995 18:01:12 EDT Runtime: 4834
18:01:12 8/7/1995 18:01:12 EDT Runtime: 80
18:01:14 ! Zippity Do Dah -- The End
18:01:14
18:01:14 % Starting BILXFR.COM[1,177777]
18:01:14 @BILXFR[1,177777]
18:01:15 BILDAY.LST
18:01:15 Billing Error Report
18:01:15 CHF
18:01:15 CSI:LSSBILLMESS
18:01:15
18:01:17 no structures
18:01:17 % IDYSNF No Information Service structure found
18:01:18 XFER.SYS
18:01:18 W256
18:01:18 IO
18:01:18 E
18:01:18 BIL32.SYS
18:01:19 W256
18:01:19 IO
18:01:19 E
18:01:23 BILCHF.PAK is ready
18:02:00
18:02:00 Ntxday %1A(11)-2, Monday, August 7, 1995 6:02:00 PM EDT (CHF=DKF)
18:02:00
18:02:00 Operator ID: Normal
18:02:00
18:02:00 Processed 0 records
18:02:00
18:02:00 NTXDAY done!
18:02:01 FRNTRN.SYS
18:02:01 W256
18:02:01 IO
18:02:01 E
18:02:02 NETFIL.SYS
18:02:02 W256
18:02:02 IO
18:02:02 E
18:02:02 NETCHF.PAK is ready
18:02:02 BILING done!
```

**@BILSAV n (n=MTA number)**

## **BILLING PROGRAMS RAN ON ONE HOST ONLY**

### **RETFIL**

- Retrieves information from BILDAY and NETDAY .PAK files
- Ran twice; to retrieve BIL or NET files
- Must run before any merge programs

### **ISDMRG**

- Merges CIS billing files ISDX.\*
- Creates new ISDADD.DAT
- Copy ISDX.xxx files to MIC:

### **CISMRG**

- Merges the day's CIS billing data with the month-to-date data
- Writes newly merged data to output magtape
- CISMRG creates new BILADD.DAT, LINISD.DAT

### **BISMRG**

- Merges the day's BIS billing data with the month-to-date data
- Writes newly merged data to output magtape
- BISMRG creates new MTDX.DAT[1,177777]

### **NTXMRG**

- Uses data from .PAK files created by NTXDAY
- Merges the day's Network billing data with the month-to-date network data
- Writes newly merged data to output magtape
- Output files are copied to the accounting system

### **POSMRG**

- Merges Network Point of Sales billing data

• **Example of billing that is run on one system:**

[Session log begun at 4:58 PM EDT Wednesday, April 29, 1998]  
[Job 52 on GHH, [1,2] on T75ATR]

```
16:58:09
16:58:09 OK
16:58:09 .set noecho
16:58:09
16:58:09 OK
16:58:34 @BIL:BCLEAN TUE
16:58:34
16:58:34 OK
16:58:34 .set no output
16:58:34 % BCLEAN - Day to clean up is TUE. OK?
16:58:35 % BCLEAN - Cleaning up systems CHA-CHZ,DHA-DHZ,EHA-EHW,EHY,EHZ,FHA-
FHZ,BHA-BHZ,GHA-GHZ,HHA-HHZ,IHA-IHZ,JHA-JHZ,KHA-KHZ,LHA-LHZ. OK?
17:01:41 % BCLDLT - BCLEAN Deleting BIL:[1,2000]CISDAY.MAS
17:01:41 % BCLDLT - BCLEAN Deleting BIL:[1,2000]DAYXX.MAS
17:01:41 % BCLDLT - BCLEAN Deleting BIL:[1,2000]NTXDAY.MAS
17:01:41 % BCLDLT - BCLEAN Deleting BIL:[1,2000]FRNTRN.DAY
17:01:41 % BCLDLT - BCLEAN Deleting BACKUP:[1,2]BILADD.TUE
17:01:42 % BCLDLT - BCLEAN Deleting BACKUP:[1,2]ISDADD.TUE
17:01:42 % BCLDLT - BCLEAN Deleting BACKUP:[1,2]MISSLE.TUE
17:01:42
17:01:42 ? LISNFN - No file named DSK:*.TUE[1,1000]
17:01:42
17:01:42 % BCLEAN - Completed !!
17:01:42
17:01:42 OK
19:37:01 @BIL:BILDIR TED
19:37:02
19:37:02 OK
19:37:02 .set noecho
19:37:02
19:37:02 OK
19:37:02 .DIR/A/B/L/SOR:NAM TED:[1,2]???H?.PAK
19:37:02 Name Ext Prot Blocks Creation Access
Mode
19:37:02
19:37:02 TED:
19:37:02 BILBHA PAK (3) 41441 18:11 29-Apr-98 29-Apr-98 17
19:37:02 BILBHB PAK (3) 28963 18:06 29-Apr-98 29-Apr-98 17
19:37:02 BILBHC PAK (3) 31198 18:06 29-Apr-98 29-Apr-98 17
19:37:02 BILBHD PAK (3) 57530 18:07 29-Apr-98 29-Apr-98 17
19:37:02 BILBHE PAK (3) 24065 18:05 29-Apr-98 29-Apr-98 17
19:37:02 BILBHF PAK (3) 8092 18:04 29-Apr-98 29-Apr-98 17
19:37:02 BILBHG PAK (3) 38907 18:10 29-Apr-98 29-Apr-98 17
19:37:02 BILBHH PAK (3) 7675 18:02 29-Apr-98 29-Apr-98 17
19:37:02 BILBHI PAK (3) 15982 18:02 29-Apr-98 29-Apr-98 17
19:37:02 BILBHJ PAK (3) 8949 18:03 29-Apr-98 29-Apr-98 17
19:37:02 BILBHK PAK (3) 9654 18:03 29-Apr-98 29-Apr-98 17
19:37:02 BILBHL PAK (3) 44586 18:10 29-Apr-98 29-Apr-98 17
19:37:02 BILBHM PAK (3) 6427 18:02 29-Apr-98 29-Apr-98 17
19:37:02 BILBHN PAK (3) 7636 18:02 29-Apr-98 29-Apr-98 17
19:37:02 BILBHO PAK (3) 12840 18:04 29-Apr-98 29-Apr-98 17
19:37:02 BILBHP PAK (3) 12649 18:03 29-Apr-98 29-Apr-98 17
19:37:02 BILBHQ PAK (3) 7495 18:02 29-Apr-98 29-Apr-98 17
19:37:02 BILBHR PAK (3) 6741 18:02 29-Apr-98 29-Apr-98 17
19:37:02 BILBHS PAK (3) 9670 18:02 29-Apr-98 29-Apr-98 17
19:37:02 BILBHT PAK (3) 29130 18:05 29-Apr-98 29-Apr-98 17
19:37:02 BILBHU PAK (3) 10095 18:05 29-Apr-98 29-Apr-98 17
```

19:37:02	BILBHV	PAK	(3)	23926	18:07	29-Apr-98	29-Apr-98	17
19:37:02	BILBHW	PAK	(3)	7838	18:03	29-Apr-98	29-Apr-98	17
19:37:02	BILCHA	PAK	(3)	29287	18:12	29-Apr-98	29-Apr-98	17
19:37:02	BILCHB	PAK	(3)	25706	18:05	29-Apr-98	29-Apr-98	17
19:37:02	BILCHC	PAK	(3)	39174	18:13	29-Apr-98	29-Apr-98	17
19:37:02	BILCHD	PAK	(3)	26830	18:08	29-Apr-98	29-Apr-98	17
19:37:02	BILCHE	PAK	(3)	15676	18:12	29-Apr-98	29-Apr-98	17
19:37:02	BILCHF	PAK	(3)	5303	18:13	29-Apr-98	29-Apr-98	17
19:37:02	BILCHG	PAK	(3)	26007	18:11	29-Apr-98	29-Apr-98	17
19:37:02	BILCHH	PAK	(3)	12806	18:13	29-Apr-98	29-Apr-98	17
19:37:02	BILCHI	PAK	(3)	26015	18:08	29-Apr-98	29-Apr-98	17
19:37:02	BILCHJ	PAK	(3)	28509	18:09	29-Apr-98	29-Apr-98	17
19:37:02	BILCHK	PAK	(3)	14272	18:07	29-Apr-98	29-Apr-98	17
19:37:02	BILCHL	PAK	(3)	24680	18:09	29-Apr-98	29-Apr-98	17
19:37:02	BILCHM	PAK	(3)	41484	18:10	29-Apr-98	29-Apr-98	17
19:37:02	BILCHN	PAK	(3)	23628	18:06	29-Apr-98	29-Apr-98	17
19:37:02	BILCHO	PAK	(3)	33407	18:07	29-Apr-98	29-Apr-98	17
19:37:02	BILCHP	PAK	(3)	24984	18:05	29-Apr-98	29-Apr-98	17
19:37:02	BILCHQ	PAK	(3)	25189	18:07	29-Apr-98	29-Apr-98	17
19:37:02	BILCHR	PAK	(3)	32240	18:07	29-Apr-98	29-Apr-98	17
19:37:02	BILCHS	PAK	(3)	21497	18:09	29-Apr-98	29-Apr-98	17
19:37:02	BILCHT	PAK	(3)	8093	18:02	29-Apr-98	29-Apr-98	17
19:37:02	BILCHU	PAK	(3)	21493	18:13	29-Apr-98	29-Apr-98	17
19:37:02	BILCHV	PAK	(3)	32885	18:11	29-Apr-98	29-Apr-98	17
19:37:02	BILDHA	PAK	(3)	20234	18:10	29-Apr-98	29-Apr-98	17
19:37:02	BILDHB	PAK	(3)	35753	18:31	29-Apr-98	29-Apr-98	17
19:37:02	BILDHC	PAK	(3)	22712	18:15	29-Apr-98	29-Apr-98	17
19:37:02	BILDHD	PAK	(3)	29897	18:11	29-Apr-98	29-Apr-98	17
19:37:02	BILDHE	PAK	(3)	30260	18:08	29-Apr-98	29-Apr-98	17
19:37:02	BILDHF	PAK	(3)	27691	18:09	29-Apr-98	29-Apr-98	17
19:37:02	BILDHG	PAK	(3)	27099	18:11	29-Apr-98	29-Apr-98	17
19:37:02	BILDHH	PAK	(3)	13531	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILDHI	PAK	(3)	27894	18:14	29-Apr-98	29-Apr-98	17
19:37:03	BILDHJ	PAK	(3)	31314	18:10	29-Apr-98	29-Apr-98	17
19:37:03								
19:37:03	Name	Ext	Prot	Blocks	Creation		Access	
Mode								
19:37:03								
19:37:03	BILDHK	PAK	(3)	4708	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILDHL	PAK	(3)	20812	18:06	29-Apr-98	29-Apr-98	17
19:37:03	BILDHM	PAK	(3)	27790	18:08	29-Apr-98	29-Apr-98	17
19:37:03	BILDHN	PAK	(3)	23361	18:13	29-Apr-98	29-Apr-98	17
19:37:03	BILDHO	PAK	(3)	31528	18:08	29-Apr-98	29-Apr-98	17
19:37:03	BILDHP	PAK	(3)	4689	18:01	29-Apr-98	29-Apr-98	17
19:37:03	BILDHQ	PAK	(3)	21149	18:08	29-Apr-98	29-Apr-98	17
19:37:03	BILDHR	PAK	(3)	31591	18:10	29-Apr-98	29-Apr-98	17
19:37:03	BILDHS	PAK	(3)	41045	18:12	29-Apr-98	29-Apr-98	17
19:37:03	BILDHT	PAK	(3)	35130	18:11	29-Apr-98	29-Apr-98	17
19:37:03	BILDHU	PAK	(3)	6359	18:34	29-Apr-98	29-Apr-98	17
19:37:03	BILDHV	PAK	(3)	2896	18:01	29-Apr-98	29-Apr-98	17
19:37:03	BILEHA	PAK	(3)	18348	18:05	29-Apr-98	29-Apr-98	17
19:37:03	BILEHB	PAK	(3)	56884	18:17	29-Apr-98	29-Apr-98	17
19:37:03	BILEHC	PAK	(3)	27925	18:06	29-Apr-98	29-Apr-98	17
19:37:03	BILEHD	PAK	(3)	35819	18:06	29-Apr-98	29-Apr-98	17
19:37:03	BILEHE	PAK	(3)	98060	18:37	29-Apr-98	29-Apr-98	17
19:37:03	BILEHF	PAK	(3)	30335	18:10	29-Apr-98	29-Apr-98	17
19:37:03	BILEHG	PAK	(3)	11648	18:01	29-Apr-98	29-Apr-98	17
19:37:03	BILEHH	PAK	(3)	38256	18:07	29-Apr-98	29-Apr-98	17
19:37:03	BILEHI	PAK	(3)	25571	18:05	29-Apr-98	29-Apr-98	17
19:37:03	BILEHJ	PAK	(3)	32220	18:05	29-Apr-98	29-Apr-98	17
19:37:03	BILEHK	PAK	(3)	109474	18:39	29-Apr-98	29-Apr-98	17

19:37:03	BILEHL	PAK	(3)	64780	18:25	29-Apr-98	29-Apr-98	17
19:37:03	BILEHM	PAK	(3)	11749	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILEHN	PAK	(3)	17981	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILEHO	PAK	(3)	35805	18:07	29-Apr-98	29-Apr-98	17
19:37:03	BILEHP	PAK	(3)	37488	18:05	29-Apr-98	29-Apr-98	17
19:37:03	BILEHQ	PAK	(3)	86566	18:38	29-Apr-98	29-Apr-98	17
19:37:03	BILEHR	PAK	(3)	9948	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILEHS	PAK	(3)	15059	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILEHT	PAK	(3)	14859	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILEHU	PAK	(3)	30942	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILEHV	PAK	(3)	48114	18:05	29-Apr-98	29-Apr-98	17
19:37:03	BILFHA	PAK	(3)	4877	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHB	PAK	(3)	3478	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILFHC	PAK	(3)	7812	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHD	PAK	(3)	9846	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHE	PAK	(3)	7200	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILFHF	PAK	(3)	3875	18:01	29-Apr-98	29-Apr-98	17
19:37:03	BILFHG	PAK	(3)	3573	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHH	PAK	(3)	10007	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILFHJ	PAK	(3)	7662	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHK	PAK	(3)	4649	18:01	29-Apr-98	29-Apr-98	17
19:37:03	BILFHL	PAK	(3)	3837	18:01	29-Apr-98	29-Apr-98	17
19:37:03	BILFHM	PAK	(3)	12121	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHN	PAK	(3)	7682	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHO	PAK	(3)	9718	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHP	PAK	(3)	8243	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHQ	PAK	(3)	18205	18:07	29-Apr-98	29-Apr-98	17
19:37:03	BILFHR	PAK	(3)	7019	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILFHS	PAK	(3)	3467	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILFHT	PAK	(3)	10847	18:07	29-Apr-98	29-Apr-98	17
19:37:03	BILFHV	PAK	(3)	11407	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILFHW	PAK	(3)	4497	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILGHD	PAK	(3)	30074	18:08	29-Apr-98	29-Apr-98	17
19:37:03								
19:37:03	Name	Ext	Prot	Blocks		Creation	Access	
Mode								
19:37:03								
19:37:03	BILGHE	PAK	(3)	13713	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILGHF	PAK	(3)	10489	18:07	29-Apr-98	29-Apr-98	17
19:37:03	BILGHG	PAK	(3)	5472	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILGHH	PAK	(3)	7844	18:01	29-Apr-98	29-Apr-98	16
19:37:03	BILGHI	PAK	(3)	19415	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILGHJ	PAK	(3)	20392	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILGHK	PAK	(3)	11201	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILGHL	PAK	(3)	14226	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILGHM	PAK	(3)	42293	18:12	29-Apr-98	29-Apr-98	17
19:37:03	BILGHN	PAK	(3)	17401	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILGHO	PAK	(3)	34567	18:04	29-Apr-98	29-Apr-98	17
19:37:03	BILGHP	PAK	(3)	38582	18:07	29-Apr-98	29-Apr-98	17
19:37:03	BILGHQ	PAK	(3)	8041	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILGHR	PAK	(3)	8470	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILGHS	PAK	(3)	17593	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILGHT	PAK	(3)	23140	18:03	29-Apr-98	29-Apr-98	17
19:37:03	BILGHU	PAK	(3)	7113	18:02	29-Apr-98	29-Apr-98	17
19:37:03	BILGHV	PAK	(3)	33988	18:06	29-Apr-98	29-Apr-98	17
19:37:03	BILGHW	PAK	(3)	28293	18:04	29-Apr-98	29-Apr-98	17
19:37:03	BILHHA	PAK	(3)	14285	18:05	29-Apr-98	29-Apr-98	17
19:37:03	BILHHB	PAK	(3)	3735	18:01	29-Apr-98	29-Apr-98	17
19:37:04	BILHHG	PAK	(3)	4129	18:02	29-Apr-98	29-Apr-98	17
19:37:04	BILHHH	PAK	(3)	3200	18:01	29-Apr-98	29-Apr-98	17
19:37:04	BILHHI	PAK	(3)	3282	18:01	29-Apr-98	29-Apr-98	17

19:37:04	BILHHJ	PAK	(3)	7913	18:02	29-Apr-98	29-Apr-98	17
19:37:04	BILHHK	PAK	(3)	3134	18:01	29-Apr-98	29-Apr-98	17
19:37:04	BILHHL	PAK	(3)	6594	18:02	29-Apr-98	29-Apr-98	17
19:37:04	BILHHM	PAK	(3)	8130	18:02	29-Apr-98	29-Apr-98	17
19:37:04	BILHHN	PAK	(3)	3320	18:04	29-Apr-98	29-Apr-98	17
19:37:04	BILHHO	PAK	(3)	9625	18:03	29-Apr-98	29-Apr-98	17
19:37:04	BILHHP	PAK	(3)	2844	18:01	29-Apr-98	29-Apr-98	17
19:37:04	BILHHQ	PAK	(3)	8450	18:02	29-Apr-98	29-Apr-98	17
19:37:04	BILHHR	PAK	(3)	4934	18:02	29-Apr-98	29-Apr-98	17
19:37:04	BILHHS	PAK	(3)	8337	18:02	29-Apr-98	29-Apr-98	17
19:37:04	BILHHT	PAK	(3)	3560	18:01	29-Apr-98	29-Apr-98	17
19:37:04	BILHHU	PAK	(3)	7749	18:02	29-Apr-98	29-Apr-98	17
19:37:04	BILHHV	PAK	(3)	3998	18:02	29-Apr-98	29-Apr-98	17
19:37:05	BILHHW	PAK	(3)	3732	18:01	29-Apr-98	29-Apr-98	17
19:37:05	BILIHA	PAK	(3)	11536	18:02	29-Apr-98	29-Apr-98	17
19:37:05	BILIHb	PAK	(3)	9128	18:01	29-Apr-98	29-Apr-98	17
19:37:05	BILIHc	PAK	(3)	18539	18:03	29-Apr-98	29-Apr-98	17
19:37:05	BILIHd	PAK	(3)	23585	18:03	29-Apr-98	29-Apr-98	17
19:37:05	BILIHe	PAK	(3)	60587	18:34	29-Apr-98	29-Apr-98	17
19:37:05	BILIHf	PAK	(3)	83698	18:46	29-Apr-98	29-Apr-98	17
19:37:05	BILIHg	PAK	(3)	52756	18:32	29-Apr-98	29-Apr-98	17
19:37:05	BILIHh	PAK	(3)	11457	18:02	29-Apr-98	29-Apr-98	17
19:37:05	BILIHl	PAK	(3)	59038	18:38	29-Apr-98	29-Apr-98	17
19:37:05	BILIHj	PAK	(3)	29072	18:05	29-Apr-98	29-Apr-98	17
19:37:05	BILIHk	PAK	(3)	91721	18:34	29-Apr-98	29-Apr-98	17
19:37:05	BILIHl	PAK	(3)	16487	18:03	29-Apr-98	29-Apr-98	17
19:37:05	BILIHm	PAK	(3)	19052	18:04	29-Apr-98	29-Apr-98	17
19:37:05	BILIHn	PAK	(3)	54467	18:13	29-Apr-98	29-Apr-98	17
19:37:05	BILIHo	PAK	(3)	23608	18:06	29-Apr-98	29-Apr-98	17
19:37:06	BILIHp	PAK	(3)	52913	18:08	29-Apr-98	29-Apr-98	17
19:37:06	BILIHq	PAK	(3)	4226	18:01	29-Apr-98	29-Apr-98	17
19:37:06	BILIHr	PAK	(3)	15608	18:03	29-Apr-98	29-Apr-98	17
19:37:06								
19:37:06	Name	Ext	Prot	Blocks	Creation		Access	
19:37:06	Mode							
19:37:06								
19:37:06	BILIHS	PAK	(3)	67912	18:48	29-Apr-98	29-Apr-98	17
19:37:06	BILIHT	PAK	(3)	35257	18:03	29-Apr-98	29-Apr-98	17
19:37:06	BILIHU	PAK	(3)	19914	18:02	29-Apr-98	29-Apr-98	17
19:37:06	BILIHV	PAK	(3)	21826	18:03	29-Apr-98	29-Apr-98	17
19:37:06	BILIHW	PAK	(3)	45261	18:05	29-Apr-98	29-Apr-98	17
19:37:06	BILJHA	PAK	(3)	21260	18:04	29-Apr-98	29-Apr-98	17
19:37:06	BILJHB	PAK	(3)	73740	18:41	29-Apr-98	29-Apr-98	17
19:37:06	BILJHC	PAK	(3)	32630	18:04	29-Apr-98	29-Apr-98	17
19:37:06	BILJHD	PAK	(3)	26170	18:05	29-Apr-98	29-Apr-98	17
19:37:06	BILJHE	PAK	(3)	63726	18:37	29-Apr-98	29-Apr-98	17
19:37:06	BILJHF	PAK	(3)	77572	18:44	29-Apr-98	29-Apr-98	17
19:37:06	BILJHG	PAK	(3)	17798	18:05	29-Apr-98	29-Apr-98	17
19:37:06	BILJHH	PAK	(3)	8360	18:01	29-Apr-98	29-Apr-98	17
19:37:07	BILJHI	PAK	(3)	6375	18:02	29-Apr-98	29-Apr-98	17
19:37:07	BILJHJ	PAK	(3)	7159	18:03	29-Apr-98	29-Apr-98	17
19:37:07	BILJHK	PAK	(3)	47089	18:29	29-Apr-98	29-Apr-98	17
19:37:07	BILJHL	PAK	(3)	56361	18:32	29-Apr-98	29-Apr-98	17
19:37:07	BILJHM	PAK	(3)	73574	18:35	29-Apr-98	29-Apr-98	17
19:37:07	BILJHN	PAK	(3)	22533	18:05	29-Apr-98	29-Apr-98	17
19:37:07	BILJHO	PAK	(3)	21246	18:06	29-Apr-98	29-Apr-98	17
19:37:07	BILJHP	PAK	(3)	74525	18:42	29-Apr-98	29-Apr-98	17
19:37:07	BILJHQ	PAK	(3)	15910	18:02	29-Apr-98	29-Apr-98	17
19:37:07	BILJHR	PAK	(3)	57155	18:31	29-Apr-98	29-Apr-98	17
19:37:07	BILJHS	PAK	(3)	3539	18:01	29-Apr-98	29-Apr-98	17
19:37:07	BILJHT	PAK	(3)	76071	18:36	29-Apr-98	29-Apr-98	17

19:37:07	BILJHU	PAK	(3)	94575	19:11	29-Apr-98	29-Apr-98	17
19:37:07	BILJHV	PAK	(3)	4730	18:02	29-Apr-98	29-Apr-98	17
19:37:07	BILJHW	PAK	(3)	4182	18:01	29-Apr-98	29-Apr-98	17
19:37:07	NETBHA	PAK	(3)	5201	18:11	29-Apr-98	29-Apr-98	17
19:37:08	NETBHB	PAK	(3)	6013	18:07	29-Apr-98	29-Apr-98	17
19:37:08	NETBHC	PAK	(3)	4161	18:06	29-Apr-98	29-Apr-98	17
19:37:08	NETBHD	PAK	(3)	5338	18:07	29-Apr-98	29-Apr-98	17
19:37:08	NETBHE	PAK	(3)	5996	18:05	29-Apr-98	29-Apr-98	17
19:37:08	NETBHF	PAK	(3)	4	18:04	29-Apr-98	29-Apr-98	17
19:37:08	NETBHG	PAK	(3)	4	18:10	29-Apr-98	29-Apr-98	17
19:37:08	NETBHH	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:08	NETBHI	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:08	NETBHJ	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:08	NETBHK	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:08	NETBHL	PAK	(3)	4	18:10	29-Apr-98	29-Apr-98	17
19:37:08	NETBHM	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:08	NETBHN	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:08	NETBHO	PAK	(3)	4	18:04	29-Apr-98	29-Apr-98	17
19:37:08	NETBHP	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:08	NETBHQ	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:09	NETBHR	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:09	NETBHS	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:09	NETBHT	PAK	(3)	4	18:05	29-Apr-98	29-Apr-98	17
19:37:09	NETBHU	PAK	(3)	4	18:05	29-Apr-98	29-Apr-98	17
19:37:09	NETBHV	PAK	(3)	4	18:07	29-Apr-98	29-Apr-98	17
19:37:09	NETBHW	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:09	NETCHA	PAK	(3)	7060	18:12	29-Apr-98	29-Apr-98	17
19:37:09	NETCHB	PAK	(3)	7799	18:06	29-Apr-98	29-Apr-98	17
19:37:09	NETCHC	PAK	(3)	16133	18:15	29-Apr-98	29-Apr-98	17
19:37:09	NETCHD	PAK	(3)	5281	18:09	29-Apr-98	29-Apr-98	17
19:37:09	NETCHE	PAK	(3)	6412	18:12	29-Apr-98	29-Apr-98	17
19:37:09								
19:37:09	Name	Ext	Prot	Blocks	Creation		Access	
19:37:09	Mode							
19:37:09	NETCHF	PAK	(3)	4	18:13	29-Apr-98	29-Apr-98	17
19:37:09	NETCHG	PAK	(3)	7774	18:12	29-Apr-98	29-Apr-98	17
19:37:09	NETCHH	PAK	(3)	2295	18:14	29-Apr-98	29-Apr-98	17
19:37:09	NETCHI	PAK	(3)	5714	18:09	29-Apr-98	29-Apr-98	17
19:37:10	NETCHJ	PAK	(3)	10853	18:10	29-Apr-98	29-Apr-98	17
19:37:10	NETCHK	PAK	(3)	2774	18:07	29-Apr-98	29-Apr-98	17
19:37:10	NETCHL	PAK	(3)	8336	18:10	29-Apr-98	29-Apr-98	17
19:37:10	NETCHM	PAK	(3)	20534	18:12	29-Apr-98	29-Apr-98	17
19:37:10	NETCHN	PAK	(3)	4635	18:07	29-Apr-98	29-Apr-98	17
19:37:10	NETCHO	PAK	(3)	5135	18:08	29-Apr-98	29-Apr-98	17
19:37:10	NETCHP	PAK	(3)	6192	18:05	29-Apr-98	29-Apr-98	17
19:37:10	NETCHQ	PAK	(3)	7911	18:08	29-Apr-98	29-Apr-98	17
19:37:10	NETCHR	PAK	(3)	11028	18:08	29-Apr-98	29-Apr-98	17
19:37:10	NETCHS	PAK	(3)	6134	18:09	29-Apr-98	29-Apr-98	17
19:37:10	NETCHT	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:10	NETCHU	PAK	(3)	2748	18:13	29-Apr-98	29-Apr-98	17
19:37:10	NETCHV	PAK	(3)	12752	18:13	29-Apr-98	29-Apr-98	17
19:37:10	NETDHA	PAK	(3)	5833	18:10	29-Apr-98	29-Apr-98	17
19:37:10	NETDHB	PAK	(3)	7404	18:32	29-Apr-98	29-Apr-98	17
19:37:10	NETDHC	PAK	(3)	8987	18:17	29-Apr-98	29-Apr-98	17
19:37:11	NETDHD	PAK	(3)	12502	18:12	29-Apr-98	29-Apr-98	17
19:37:11	NETDHE	PAK	(3)	8506	18:09	29-Apr-98	29-Apr-98	17
19:37:11	NETDHF	PAK	(3)	10204	18:10	29-Apr-98	29-Apr-98	17
19:37:11	NETDHG	PAK	(3)	7578	18:12	29-Apr-98	29-Apr-98	17
19:37:11	NETDHH	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:11	NETDHI	PAK	(3)	6988	18:14	29-Apr-98	29-Apr-98	17

19:37:11	NETDHJ	PAK	(3)	11671	18:12	29-Apr-98	29-Apr-98	17
19:37:11	NETDHL	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:11	NETDHL	PAK	(3)	4226	18:06	29-Apr-98	29-Apr-98	17
19:37:11	NETDHL	PAK	(3)	9884	18:09	29-Apr-98	29-Apr-98	17
19:37:11	NETDHN	PAK	(3)	6750	18:14	29-Apr-98	29-Apr-98	17
19:37:11	NETDHO	PAK	(3)	12071	18:10	29-Apr-98	29-Apr-98	17
19:37:11	NETDHP	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:11	NETDHO	PAK	(3)	5370	18:09	29-Apr-98	29-Apr-98	17
19:37:11	NETDHR	PAK	(3)	8286	18:11	29-Apr-98	29-Apr-98	17
19:37:11	NETDHS	PAK	(3)	14006	18:15	29-Apr-98	29-Apr-98	17
19:37:11	NETDHT	PAK	(3)	10242	18:12	29-Apr-98	29-Apr-98	17
19:37:12	NETDHU	PAK	(3)	4	18:35	29-Apr-98	29-Apr-98	17
19:37:12	NETDHV	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:12	NETEHA	PAK	(3)	2150	18:05	29-Apr-98	29-Apr-98	17
19:37:12	NETEHB	PAK	(3)	7030	18:18	29-Apr-98	29-Apr-98	17
19:37:12	NETEHC	PAK	(3)	10955	18:06	29-Apr-98	29-Apr-98	17
19:37:12	NETEHD	PAK	(3)	14062	18:07	29-Apr-98	29-Apr-98	17
19:37:12	NETEHE	PAK	(3)	29805	18:39	29-Apr-98	29-Apr-98	17
19:37:12	NETEHF	PAK	(3)	7259	18:11	29-Apr-98	29-Apr-98	17
19:37:12	NETEHG	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:12	NETEHH	PAK	(3)	9992	18:07	29-Apr-98	29-Apr-98	17
19:37:12	NETEHI	PAK	(3)	8113	18:06	29-Apr-98	29-Apr-98	17
19:37:12	NETEHJ	PAK	(3)	4148	18:06	29-Apr-98	29-Apr-98	17
19:37:12	NETEHK	PAK	(3)	34864	18:43	29-Apr-98	29-Apr-98	17
19:37:12	NETEHL	PAK	(3)	10778	18:26	29-Apr-98	29-Apr-98	17
19:37:12	NETEHM	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:12	NETEHN	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:13	NETEHO	PAK	(3)	4	18:07	29-Apr-98	29-Apr-98	17
19:37:13	NETEHP	PAK	(3)	4	18:05	29-Apr-98	29-Apr-98	17
19:37:13	NETEHQ	PAK	(3)	29890	18:40	29-Apr-98	29-Apr-98	17
19:37:13								
19:37:13	Name	Ext	Prot	Blocks	Creation		Access	
19:37:13	Mode							
19:37:13								
19:37:13	NETEHR	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:13	NETEHS	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:13	NETEHT	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:13	NETEHU	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:13	NETEHV	PAK	(3)	4	18:05	29-Apr-98	29-Apr-98	17
19:37:13	NETFHA	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:13	NETFHB	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:13	NETFHC	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:13	NETFHD	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:13	NETFHE	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:13	NETFHF	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:13	NETFHG	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:14	NETFHH	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:14	NETFHJ	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:14	NETFHK	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:14	NETFHL	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:14	NETFHM	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:14	NETFHN	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:14	NETFHO	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:14	NETFHP	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:14	NETFHQ	PAK	(3)	4	18:07	29-Apr-98	29-Apr-98	17
19:37:14	NETFHR	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:14	NETFHS	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:14	NETFHT	PAK	(3)	4	18:07	29-Apr-98	29-Apr-98	17
19:37:14	NETFHV	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:14	NETFHW	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:14	NETGHD	PAK	(3)	4	18:09	29-Apr-98	29-Apr-98	17

19:37:14	NETGHE	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:15	NETGHF	PAK	(3)	4	18:07	29-Apr-98	29-Apr-98	17
19:37:15	NETGHG	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:15	NETGHH	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	16
19:37:15	NETGHI	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:15	NETGHJ	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:15	NETGHK	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:15	NETGHL	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:15	NETGHM	PAK	(3)	4	18:12	29-Apr-98	29-Apr-98	17
19:37:15	NETGHN	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:15	NETGHO	PAK	(3)	4	18:04	29-Apr-98	29-Apr-98	17
19:37:15	NETGHP	PAK	(3)	4	18:07	29-Apr-98	29-Apr-98	17
19:37:15	NETGHQ	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:15	NETGHR	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:15	NETGHS	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:15	NETGHT	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:15	NETGHU	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:15	NETGHV	PAK	(3)	4	18:06	29-Apr-98	29-Apr-98	17
19:37:16	NETGHW	PAK	(3)	4	18:04	29-Apr-98	29-Apr-98	17
19:37:16	NETHHA	PAK	(3)	4	18:05	29-Apr-98	29-Apr-98	17
19:37:16	NETHHB	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:16	NETHHG	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:16	NETHHH	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:16	NETHHI	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:16	NETHHJ	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:16	NETHHK	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:17	NETHHL	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:17	NETHHM	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:17	NETHHN	PAK	(3)	4	18:04	29-Apr-98	29-Apr-98	17
19:37:17								
19:37:17	Name	Ext	Prot	Blocks	Creation		Access	
19:37:17	Mode							
19:37:17	NETHHO	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:17	NETHHP	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:17	NETHHQ	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:17	NETHHR	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:17	NETHHS	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:17	NETHHT	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:17	NETHHU	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:17	NETHHV	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:17	NETHHW	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:17	NETIHA	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:17	NETIHB	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:17	NETIHC	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:18	NETIHD	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:18	NETIHE	PAK	(3)	27351	18:36	29-Apr-98	29-Apr-98	17
19:37:18	NETIHF	PAK	(3)	28871	18:48	29-Apr-98	29-Apr-98	17
19:37:18	NETIHG	PAK	(3)	29288	18:33	29-Apr-98	29-Apr-98	17
19:37:18	NETIHH	PAK	(3)	4	18:02	29-Apr-98	29-Apr-98	17
19:37:18	NETIHI	PAK	(3)	28296	18:40	29-Apr-98	29-Apr-98	17
19:37:18	NETIHJ	PAK	(3)	4	18:05	29-Apr-98	29-Apr-98	17
19:37:18	NETIHK	PAK	(3)	26774	18:36	29-Apr-98	29-Apr-98	17
19:37:18	NETIHL	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:18	NETIHM	PAK	(3)	4	18:04	29-Apr-98	29-Apr-98	17
19:37:18	NETIHN	PAK	(3)	4	18:13	29-Apr-98	29-Apr-98	17
19:37:18	NETIHO	PAK	(3)	4	18:06	29-Apr-98	29-Apr-98	17
19:37:18	NETIHP	PAK	(3)	4	18:08	29-Apr-98	29-Apr-98	17
19:37:18	NETIHQ	PAK	(3)	4	18:01	29-Apr-98	29-Apr-98	17
19:37:18	NETIHR	PAK	(3)	4	18:03	29-Apr-98	29-Apr-98	17
19:37:18	NETIHS	PAK	(3)	27574	18:55	29-Apr-98	29-Apr-98	17

```

19:37:19 NETIHT PAK (3) 4 18:03 29-Apr-98 29-Apr-98 17
19:37:19 NETIHU PAK (3) 4 18:02 29-Apr-98 29-Apr-98 17
19:37:19 NETIHV PAK (3) 4 18:03 29-Apr-98 29-Apr-98 17
19:37:19 NETIHW PAK (3) 4 18:05 29-Apr-98 29-Apr-98 17
19:37:19 NETJHA PAK (3) 4 18:04 29-Apr-98 29-Apr-98 17
19:37:19 NETJHB PAK (3) 29166 18:44 29-Apr-98 29-Apr-98 17
19:37:19 NETJHC PAK (3) 4 18:04 29-Apr-98 29-Apr-98 17
19:37:19 NETJHD PAK (3) 4 18:05 29-Apr-98 29-Apr-98 17
19:37:19 NETJHE PAK (3) 26760 18:39 29-Apr-98 29-Apr-98 17
19:37:19 NETJHF PAK (3) 23976 18:47 29-Apr-98 29-Apr-98 17
19:37:19 NETJHG PAK (3) 4 18:05 29-Apr-98 29-Apr-98 17
19:37:19 NETJHH PAK (3) 4 18:01 29-Apr-98 29-Apr-98 17
19:37:19 NETJHI PAK (3) 4 18:02 29-Apr-98 29-Apr-98 17
19:37:19 NETJHJ PAK (3) 4 18:03 29-Apr-98 29-Apr-98 17
19:37:19 NETJHK PAK (3) 26068 18:31 29-Apr-98 29-Apr-98 17
19:37:19 NETJHL PAK (3) 27010 18:35 29-Apr-98 29-Apr-98 17
19:37:19 NETJHM PAK (3) 24888 18:37 29-Apr-98 29-Apr-98 17
19:37:20 NETJHN PAK (3) 4 18:05 29-Apr-98 29-Apr-98 17
19:37:20 NETJHO PAK (3) 4 18:06 29-Apr-98 29-Apr-98 17
19:37:20 NETJHP PAK (3) 28651 18:45 29-Apr-98 29-Apr-98 17
19:37:20 NETJHQ PAK (3) 4 18:02 29-Apr-98 29-Apr-98 17
19:37:20 NETJHR PAK (3) 27978 18:34 29-Apr-98 29-Apr-98 17
19:37:20 NETJHS PAK (3) 4 18:01 29-Apr-98 29-Apr-98 17
19:37:20 NETJHT PAK (3) 26967 18:38 29-Apr-98 29-Apr-98 17
19:37:20 NETJHU PAK (3) 27873 19:24 29-Apr-98 29-Apr-98 17
19:37:20 NETJHV PAK (3) 4 18:02 29-Apr-98 29-Apr-98 17
19:37:20 NETJHW PAK (3) 4 18:01 29-Apr-98 29-Apr-98 17
19:37:20
19:37:20 Total blocks 5684890 in 390 files
19:37:20
19:37:20 OK
19:37:20 .return
19:37:20
19:37:20 OK
19:37:34 R SPT:RETFIL
19:37:34
19:37:34 Normal daily run? Y
19:37:35 BIL or NET pack files? BIL
19:37:36 Source device: TED
19:59:40 %RTFEND - RETFIL Done!
19:59:40
19:59:40 OK
20:01:00 @BIL:MISSLE
20:01:00
20:01:00 OK
20:01:00 .set noecho
20:01:00 start of missile.com
20:01:00 CONCAT version 1A(3)
20:01:00 *BIL:[1,2000]missile.sys=BIL:[1,1001-1032]missile.sys
20:01:01 % CON - adding BIL:[1,1001]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1002]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1003]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1004]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1005]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1006]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1007]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1010]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1011]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1012]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1013]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1014]MISSLE.SYS

```

```
20:01:01 % CON - adding BIL:[1,1015]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1016]MISSLE.SYS
20:01:01 % CON - adding BIL:[1,1017]MISSLE.SYS
20:01:02 % CON - adding BIL:[1,1020]MISSLE.SYS
20:01:04 % CON - adding BIL:[1,1021]MISSLE.SYS
20:01:04 % CON - adding BIL:[1,1022]MISSLE.SYS
20:01:04 % CON - adding BIL:[1,1023]MISSLE.SYS
20:01:05 % CON - adding BIL:[1,1024]MISSLE.SYS
20:01:05 % CON - adding BIL:[1,1025]MISSLE.SYS
20:01:05 % CON - adding BIL:[1,1026]MISSLE.SYS
20:01:06 *exi
20:01:06 CONCAT version 1A(3)
20:01:06 *BIL:[1,2000]missle.sys=BIL:[1,1101-1132]missle.sys
20:01:06 % CON - adding BIL:[1,1101]MISSLE.SYS
20:01:06 % CON - adding BIL:[1,1102]MISSLE.SYS
20:01:06 % CON - adding BIL:[1,1103]MISSLE.SYS
20:01:07 % CON - adding BIL:[1,1104]MISSLE.SYS
20:01:07 % CON - adding BIL:[1,1105]MISSLE.SYS
20:01:07 % CON - adding BIL:[1,1106]MISSLE.SYS
20:01:07 % CON - adding BIL:[1,1107]MISSLE.SYS
20:01:07 % CON - adding BIL:[1,1110]MISSLE.SYS
20:01:07 % CON - adding BIL:[1,1111]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1112]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1113]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1114]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1115]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1116]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1117]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1120]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1121]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1122]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1123]MISSLE.SYS
20:01:08 % CON - adding BIL:[1,1124]MISSLE.SYS
20:01:09 % CON - adding BIL:[1,1125]MISSLE.SYS
20:01:09 % CON - adding BIL:[1,1126]MISSLE.SYS
20:01:09 *exi
20:01:09 CONCAT version 1A(3)
20:01:09 *BIL:[1,2000]missle.sys=BIL:[1,1201-1232]missle.sys
20:01:10 % CON - adding BIL:[1,1201]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1202]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1203]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1204]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1205]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1206]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1207]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1210]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1211]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1212]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1213]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1214]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1215]MISSLE.SYS
20:01:10 % CON - adding BIL:[1,1216]MISSLE.SYS
20:01:11 % CON - adding BIL:[1,1217]MISSLE.SYS
20:01:11 % CON - adding BIL:[1,1220]MISSLE.SYS
20:01:11 % CON - adding BIL:[1,1221]MISSLE.SYS
20:01:11 % CON - adding BIL:[1,1222]MISSLE.SYS
20:01:11 % CON - adding BIL:[1,1223]MISSLE.SYS
20:01:11 % CON - adding BIL:[1,1224]MISSLE.SYS
20:01:11 % CON - adding BIL:[1,1225]MISSLE.SYS
20:01:11 % CON - adding BIL:[1,1226]MISSLE.SYS
20:01:12 *exi
```

```
20:01:12 CONCAT version 1A(3)
20:01:12 *BIL: [1,2000]missile.sys=BIL: [1,1301-1332]missile.sys
20:01:13 % CON - adding BIL: [1,1301]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1302]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1303]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1304]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1305]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1306]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1307]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1310]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1312]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1313]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1314]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1315]MISSLE.SYS
20:01:13 % CON - adding BIL: [1,1316]MISSLE.SYS
20:01:14 % CON - adding BIL: [1,1317]MISSLE.SYS
20:01:14 % CON - adding BIL: [1,1320]MISSLE.SYS
20:01:14 % CON - adding BIL: [1,1321]MISSLE.SYS
20:01:14 % CON - adding BIL: [1,1322]MISSLE.SYS
20:01:14 % CON - adding BIL: [1,1323]MISSLE.SYS
20:01:14 % CON - adding BIL: [1,1324]MISSLE.SYS
20:01:14 % CON - adding BIL: [1,1326]MISSLE.SYS
20:01:14 % CON - adding BIL: [1,1327]MISSLE.SYS
20:01:15 *exi
20:01:15 CONCAT version 1A(3)
20:01:15 *BIL: [1,2000]missile.sys=BIL: [1,1401-1432]missile.sys
20:01:16 % CON - adding BIL: [1,1401]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1402]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1403]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1404]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1405]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1406]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1407]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1410]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1411]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1412]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1413]MISSLE.SYS
20:01:16 % CON - adding BIL: [1,1414]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1415]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1416]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1417]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1420]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1421]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1422]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1423]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1424]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1425]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1426]MISSLE.SYS
20:01:17 % CON - adding BIL: [1,1427]MISSLE.SYS
20:01:18 *exi
20:01:18 CONCAT version 1A(3)
20:01:18 *BIL: [1,2000]missile.sys=BIL: [1,1501-1532]missile.sys
20:01:19 % CON - adding BIL: [1,1504]MISSLE.SYS
20:01:19 % CON - adding BIL: [1,1505]MISSLE.SYS
20:01:19 % CON - adding BIL: [1,1506]MISSLE.SYS
20:01:19 % CON - adding BIL: [1,1507]MISSLE.SYS
20:01:19 % CON - adding BIL: [1,1510]MISSLE.SYS
20:01:19 % CON - adding BIL: [1,1511]MISSLE.SYS
20:01:19 % CON - adding BIL: [1,1512]MISSLE.SYS
20:01:19 % CON - adding BIL: [1,1513]MISSLE.SYS
20:01:19 % CON - adding BIL: [1,1514]MISSLE.SYS
```

```
20:01:19 % CON - adding BIL: [1,1515]MISSILE.SYS
20:01:19 % CON - adding BIL: [1,1516]MISSILE.SYS
20:01:19 % CON - adding BIL: [1,1517]MISSILE.SYS
20:01:25 % CON - adding BIL: [1,1520]MISSILE.SYS
20:01:25 % CON - adding BIL: [1,1521]MISSILE.SYS
20:01:25 % CON - adding BIL: [1,1522]MISSILE.SYS
20:01:25 % CON - adding BIL: [1,1523]MISSILE.SYS
20:01:25 % CON - adding BIL: [1,1524]MISSILE.SYS
20:01:26 % CON - adding BIL: [1,1525]MISSILE.SYS
20:01:26 % CON - adding BIL: [1,1526]MISSILE.SYS
20:01:26 % CON - adding BIL: [1,1527]MISSILE.SYS
20:01:26 *exi
20:01:26 CONCAT version 1A(3)
20:01:26 *BIL: [1,2000]missile.sys=BIL: [1,1601-1632]missile.sys
20:01:27 % CON - adding BIL: [1,1601]MISSILE.SYS
20:01:27 % CON - adding BIL: [1,1602]MISSILE.SYS
20:01:27 % CON - adding BIL: [1,1607]MISSILE.SYS
20:01:27 % CON - adding BIL: [1,1610]MISSILE.SYS
20:01:27 % CON - adding BIL: [1,1611]MISSILE.SYS
20:01:27 % CON - adding BIL: [1,1612]MISSILE.SYS
20:01:27 % CON - adding BIL: [1,1613]MISSILE.SYS
20:01:27 % CON - adding BIL: [1,1614]MISSILE.SYS
20:01:28 % CON - adding BIL: [1,1615]MISSILE.SYS
20:01:28 % CON - adding BIL: [1,1616]MISSILE.SYS
20:01:28 % CON - adding BIL: [1,1617]MISSILE.SYS
20:01:28 % CON - adding BIL: [1,1620]MISSILE.SYS
20:01:28 % CON - adding BIL: [1,1621]MISSILE.SYS
20:01:28 % CON - adding BIL: [1,1622]MISSILE.SYS
20:01:28 % CON - adding BIL: [1,1623]MISSILE.SYS
20:01:28 % CON - adding BIL: [1,1624]MISSILE.SYS
20:01:29 % CON - adding BIL: [1,1625]MISSILE.SYS
20:01:29 % CON - adding BIL: [1,1626]MISSILE.SYS
20:01:29 % CON - adding BIL: [1,1627]MISSILE.SYS
20:01:29 *exi
20:01:29 CONCAT version 1A(3)
20:01:29 *BIL: [1,2000]missile.sys=BIL: [1,1701-1732]missile.sys
20:01:30 % CON - adding BIL: [1,1701]MISSILE.SYS
20:01:30 % CON - adding BIL: [1,1702]MISSILE.SYS
20:01:30 % CON - adding BIL: [1,1703]MISSILE.SYS
20:01:30 % CON - adding BIL: [1,1704]MISSILE.SYS
20:01:31 % CON - adding BIL: [1,1705]MISSILE.SYS
20:01:31 % CON - adding BIL: [1,1706]MISSILE.SYS
20:01:31 % CON - adding BIL: [1,1707]MISSILE.SYS
20:01:31 % CON - adding BIL: [1,1710]MISSILE.SYS
20:01:31 % CON - adding BIL: [1,1711]MISSILE.SYS
20:01:31 % CON - adding BIL: [1,1712]MISSILE.SYS
20:01:31 % CON - adding BIL: [1,1713]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1714]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1715]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1716]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1717]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1720]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1721]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1722]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1723]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1724]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1725]MISSILE.SYS
20:01:32 % CON - adding BIL: [1,1726]MISSILE.SYS
20:01:33 % CON - adding BIL: [1,1727]MISSILE.SYS
20:01:33 *exi
20:01:33 CONCAT version 1A(3)
```

```
20:01:33 *BIL:[1,2000]missile.sys=BIL:[1,2001-2032]missile.sys
20:01:34 % CON - adding BIL:[1,2001]MISSILE.SYS
20:01:34 % CON - adding BIL:[1,2002]MISSILE.SYS
20:01:34 % CON - adding BIL:[1,2003]MISSILE.SYS
20:01:34 % CON - adding BIL:[1,2004]MISSILE.SYS
20:01:34 % CON - adding BIL:[1,2005]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2006]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2007]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2010]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2011]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2012]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2013]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2014]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2015]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2016]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2017]MISSILE.SYS
20:01:35 % CON - adding BIL:[1,2020]MISSILE.SYS
20:01:37 % CON - adding BIL:[1,2021]MISSILE.SYS
20:01:37 % CON - adding BIL:[1,2022]MISSILE.SYS
20:01:37 % CON - adding BIL:[1,2023]MISSILE.SYS
20:01:37 % CON - adding BIL:[1,2024]MISSILE.SYS
20:01:37 % CON - adding BIL:[1,2025]MISSILE.SYS
20:01:37 % CON - adding BIL:[1,2026]MISSILE.SYS
20:01:37 % CON - adding BIL:[1,2027]MISSILE.SYS
20:01:37 *exi
20:01:37 CONCAT version 1A(3)
20:01:37 *BIL:[1,2000]missile.sys=BIL:[1,2101-2132]missile.sys
20:01:38 ? CONNFF - no files found
20:01:38 *exi
20:01:38 CONCAT version 1A(3)
20:01:38 *BIL:[1,2000]missile.sys=BIL:[1,2201-2232]missile.sys
20:01:38 ? CONNFF - no files found
20:01:38 *exi
20:01:39
20:01:39 OK
20:01:39 .dir/1/a/b/noh BIL:[1,2]missile.Wed
20:01:39 BIL:
20:01:39 MISSLE WED (3) 1695 20:01 29-Apr-98 29-Apr-98 16
20:01:39
20:01:39 Total blocks 1695 in 1 file
20:01:39
20:01:39 OK
20:01:39 .display "end of missile.com"
20:01:39 end of missile.com
20:01:39
20:01:39 OK
20:01:39 .return
20:01:39
20:01:39 OK
20:01:44 R SPT:ISDMRG
20:01:44 Normal Daily Run (Y/N)?Y

20:01:45 ISDMRG - performing operational logging merge for 4/28/98
20:03:02 ***** File transfer request *****
20:03:02 Please transfer DSK:*. *[1,215] to MIC:*. *[321,120]
20:03:02 Starting SYS:SORT10
20:03:02
20:03:02
20:09:56 % ICMFIN - ISDCMP finished.
20:09:56 OK
20:10:17 R SPT:BISMRG
```

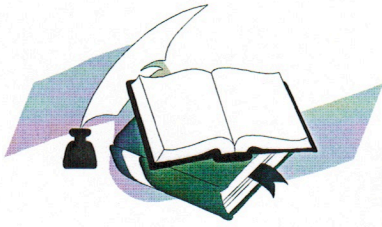
```

21:11:02
21:11:02          M87
21:11:02      BIS:WED Apr 29 Vol:1
21:11:02          Range: 0,62
21:11:02          to: 67066,6
21:11:02      Contents: MTD MAS SRC
21:11:02      [cont'd] [ ] [ ] [ ]
21:11:02
21:11:48
21:11:48 ! 1 partridge in a pear tree
21:11:48 % BMXEND - Merge Done
21:11:48
21:11:48 OK
21:27:49 R SPT:RETFIL
21:27:49
21:27:49 Normal daily run? Y
21:27:50 BIL or NET pack files? NET
21:27:54 Source device: TED
21:34:11 %RTFEND - RETFIL Done!
21:34:12
21:34:12 OK
21:55:19 R SPT:NTXMURG
21:55:19
21:55:19 Ntxmrg %1A(15)-2 Wednesday, April 29, 1998 9:55:19 PM EDT
21:55:19
21:55:19 Operator Id: GM
21:55:21
21:55:21 Normal daily run? Y
21:55:22
21:55:22 Checking and copying source files for:
21:55:22 Cha Chb Chc Chd Che Chf Chg Chh Chi Chj Chk
21:55:40 Chl Chm Chn Cho Chp Chq Chr Chs Cht Chu Chv
21:56:01 Dha Dhb Dhc Dhd Dhe Dhf Dhg Dhh Dhi Dhj Dhk
21:56:22 Dh1 Dh2 Dh3 Dh4 Dh5 Dh6 Dh7 Dh8 Dh9 Dha Dhv
21:56:42 Eha Ehb Ehc Ehd Ehe Ehf Ehg Ehh Ehi Ehj Ehk
21:57:16 Ehl Ehm Ehn Eho Ehp Ehq Ehr Ehs Eht Ehu Ehv
21:57:28 Fha Fhb Fhc Fhd Fhe Fhf Fhg Fhh Fhj Fhk Fhl
21:57:29 Fhm Fhn Fho Fhp Fhq Fhr Fhs Fht Fhv Fhw Bha
21:57:33 Bhb Bhc Bhd Bhe Bhf Bhg Bhh Bhi Bhj Bhk Bh1
21:57:40 Bhm Bhn Bho Bhp Bhq Bhr Bhs Bht Bhu Bhv Bhw
21:57:42 Ghd Ghe Ghe Ghf Ghg Ghh Ghi Ghj Ghk Gh1 Ghm Ghn
21:57:44 Gho Ghp Ghq Ghr Ghs Ght Ghu Ghv Ghw Hha Hhb
21:57:46 Hhg Hhh Hhi Hhj Hhk Hhl Hhm Hhn Hho Hhp Hhq
21:57:48 Hhr Hhs Hht Hhu Hhv Hhw Iha Ihb Ihc Ihd Ihe
21:57:57 Ihf Ihg Ihh Ihi Ihj Ihk Ihl Ihm Ihn Iho Ihp
21:58:27 Ihq Ihr Ihs Iht Ihu Ihv Ihw Jha Jhb Jhc Jhd
21:58:43 Jhe Jhf Jhg Jhh Jhi Jhj Jhk Jhl Jhm Jhn Jho
21:59:16 Jhp Jhq Jhr Jhs Jht Jhu Jhv Jhw
21:59:46
21:59:46 Generating NTXDAY.MAS [1,2000] using:
21:59:46 Cha Chb Chc Chd Che Chf Chg Chh Chi Chj Chk
21:59:47 Chl Chm Chn Cho Chp Chq Chr Chs Cht Chu Chv
21:59:48 Dha Dhb Dhc Dhd Dhe Dhf Dhg Dhh Dhi Dhj Dhk
21:59:48 Dh1 Dh2 Dh3 Dh4 Dh5 Dh6 Dh7 Dh8 Dh9 Dha Dhv
21:59:49 Eha Ehb Ehc Ehd Ehe Ehf Ehg Ehh Ehi Ehj Ehk
21:59:49 Ehl Ehm Ehn Eho Ehp Ehq Ehr Ehs Eht Ehu Ehv
21:59:49 Fha Fhb Fhc Fhd Fhe Fhf Fhg Fhh Fhj Fhk Fhl
21:59:49 Fhm Fhn Fho Fhp Fhq Fhr Fhs Fht Fhv Fhw Bha
21:59:50 Bhb Bhc Bhd Bhe Bhf Bhg Bhh Bhi Bhj Bhk Bh1
21:59:50 Bhm Bhn Bho Bhp Bhq Bhr Bhs Bht Bhu Bhv Bhw
21:59:50 Ghd Ghe Ghe Ghf Ghg Ghh Ghi Ghj Ghk Gh1 Ghm Ghn

```

```
21:59:50 Gho Ghp Ghq Ghr Ghs Ght Ghu Ghv Ghw Hha Hhb
21:59:50 Hhg Hhh Hhi Hhj Hhk Hhl Hhm Hhn Hho Hhp Hhq
21:59:50 Hhr Hhs Hht Hhu Hhv Hhw Iha Ihb Ihc Ihd Ihe
21:59:50 Ihf Ihg Ihh Ihi Ihj Ihk Ihl Ihm Ihn Iho Ihp
21:59:50 Ihq Ihr Ihs Iht Ihu Ihv Ihw Jha Jhb Jhc Jhd
21:59:50 Jhe Jhf Jhg Jhh Jhi Jhj Jhk Jhl Jhm Jhn Jho
21:59:50 Jhp Jhq Jhr Jhs Jht Jhu Jhv Jhw
22:31:32
22:31:32 Month data span: Wednesday, April 1, 1998 12:00 AM EST
22:31:32 to: Thursday, April 30, 1998 11:59 PM EDT

22:31:32 NTM input tape(s) (first = #N114) on MZB40
22:31:43 NTM output tape(s) (first = #N121) on MZB30
22:31:48
22:31:49 % Output is not a billing tape, making a new one? Y
22:31:51 are you sure you mean "yes"? Y
22:31:52 New tape written "n" times = 0
22:31:54 ! Merging month file
23:56:23
23:56:23 ! Label output tape:
23:56:23
23:56:23 N121
23:56:23 NET:WED Apr 29 Vol:1
23:56:23 Range: 100000
23:56:23 to: CISPPP
23:56:23 Contents: MTD
23:56:23 [cont'd] [ ] [ ] [ ]
23:56:23
23:56:23 % NTMWFN Waiting for next output tape (#N122) on MZB30...thank you
23:59:23 % Output is not a billing tape, making a new one? Y
23:59:28 are you sure you mean "yes"? Y
23:59:29 New tape written "n" times = 0
00:03:25 % NTMWFN Waiting for next input tape (#N115) on MZB40...thank you
01:21:09
01:21:09 ! Label output tape:
01:21:09
01:21:09 N122
01:21:09 NET:WED Apr 29 Vol:2
01:21:09 Range: CISPPP
01:21:09 to: HILFF2
01:21:09 Contents: MTD
01:21:09 [cont'd] [x] [ ] [ ]
01:21:09
01:21:09 % NTMWFN Waiting for next output tape (#N123) on MZB30...thank you
01:23:12 % Output is not a billing tape, making a new one? Y
01:23:20 are you mean "yes"? Y
01:23:20 New tape written "n" times = 0
01:30:34 % NTMWFN Waiting for next input tape (#N116) on MZB40...thank you
02:33:06
02:33:06 ! Label output tape:
02:33:06
02:33:06 N123
02:33:06 NET:WED Apr 29 Vol:3
02:33:06 Range: HILFF2
02:33:06 to: JUM1
02:33:06 Contents: MTD
02:33:06 [cont'd] [x] [ ] [ ]
02:33:06
02:33:06 % NTMWFN Waiting for next output tape (#N124) on MZB30...thank you
02:34:54 % Output is not a billing tape, making a new one? Y
02:34:57 are you sure you mean "yes"? Y
```



# **Review**

## **1. Billing programs that run on every system**

- Billing for disk storage with BLOCKS
- Collecting billing data using BILDAY
- Collecting some CIS data using ISDDAY
- Collecting network billing data using NTXDAY
- .PAK files from billing programs

## **2. Billing programs run in Arlington only**

- Retrieving data from .PAK files using RETFIL
- Merging network billing data using NTXMRG and POSMRG
- Merging CIS data using ISDMRG
- Merging BIS billing data using BISMGRG
- Merging CIS billing data using CISMGRG

## **3. IPX billing on Saturdays and end-of-month.**

- Creating IPX billing files with IPXBIL
- Gathering the billing files with @SPLRSV



## Examination Six

1. What PPN is used to keep all live billing data?
2. Explain briefly what the program BLOCKS does?
3. What billing programs are run on every system?
4. If you receive an error message during a billing program, what should you do?
5. What files does the program RETFIL use?
6. What file is created by BISMURG that is copied around to all systems?
7. What does DUPBIL do?

## **DISK PROBLEMS**

### **LOW FREE STORAGE**

- Check with SYS F or SYSDIS
- Delete unnecessary disk storage from [1,2] and [131,1]
- Save and delete [3,6]\*.?N? (done files) to tape and hold the tape for 24 hours
- (Primary structure) Check, move, and delete any crash files in [340,341]
  - **DIR/A/B/L DSK:[340,341]CHA\*.EXE**
- (Primary structure) Save and delete [1,214]BAKBIL.??? files (**ask supervisor**)
- (IS?,IT?,IU?) delete [1,215]ISDX.<dow>
- If storage is still low, R SPT:STORAG and contact a CRS
- NITELY - Used to clean up files from free users areas
  - Always run after daily FILSAVs
  - Creates NITELY.RPT[1,300]

### **DISK ERRORS**

- SYS P
- ERRDMP
- Generates report of system error conditions
  - Uses [1,200]ERROR.SYS
  - Files created are ERRDMP.LST, ERRLOG.LST
  - Runs automatically everyday at 18:00
  - Running the program manually using @ERRDMP or @CHKOUT
  - Examples:

```
R SPT:ERRDMP
*@DAY/SINCE:1300
OK
```

```

@CHKOUT DZA0
!13:47:53(OPR)
OK
.set echo
OK
.SET LOG CHKOUT.LOG
OK
.SYSTAT P=DZA0
SYSTAT V3A(233) 13:48 EST 13-Feb-89
Disk Performance Statistics:
name          free          uid          brc          bwc          drc          dwc          mrc
DKL  100675
DZA0  100675      J12766 0A      62186  777      30733  7245  8864
name          mwc          nsr          nsr          xrr          xrw          msc          usc
DKL
DZA0          6746          0          0          0          0          9537  6423
name          cde          cdp          sde          hde          spe          hpe          sse
DKL
DZA0          0          0          0          0          0          0          0
name          hse          sdv          hdv          rbe          ste          coe          doe
DKL
DZA0          0          0          0          0          0          0          221
name          wse          shd          hhd          dvh          dve          cub          cue
DKL
DZA0          0          0          0          0          0          1          1
name          ure
DKL
DZA0          0

OK
.SET NOLOG
OK
.R SPT:ERRDMP
**DSK:CHKOUT.LST=/SUMMARY/IGNORE/UNITS:DZA0-
#*/LOC:(RER,CDE,DAT,SDV,DEV,SHG,HNG,SER,SEK)/LEV:3/PAGELIMIT:DISK:150  OK
.PRINTNH/R/D/18M/LOC:COL CHKOUT.LOG,CHKOUT.LST
Line1: *SYS P and ERRDMP
Line2: *for: DZA0
Line3: *System: CHL
Line4: *bring to Computer Room
OK
.return
OK

```

- Fill out an ESR and/or contact ISG.
- Make an entry in the hardware section of the report

## **DISK ERROR UTILITY PROGRAMS**

- **SLOVER**

SLOVER is a program which allows the monitor to perform bad block marking for disk drives. This program is primarily run after a drive has been formatted/tested using one of the disk drive formatter programs.

- **EXAMPLE:**

```
R SPT:SLOVER
slover version 0(1),0
SLOVER> DZA0:
logical blocks in decimal
first=97686 last=97686 size=1
SLOVER> <CR>
OK
```

- **DSKRAT**

DSKRAT is the disk performance rating program. When a crash or monitor error occurs, it is possible that disk storage will be lost. That is, specific clusters will be marked as in-use and will not be contained within any file. An example of a situation that could cause this is a crash during the processing of a DELETE command. If the directory (UFD) is cleared of the reference to the file, and the SAT (storage allocation table) file was not yet updated to indicate that the clusters are now free, they would be "lost". Another example of disks in an improper state is a cluster belonging to more than one file (multiply-used), or a cluster belonging to a file but not marked in the SAT file as "in use" (free).

Since DSKRAT is endeavoring to determine the instantaneous status of the structure, it is necessary to insure that no files on that structure are accessed or written during the run time of DSKRAT. However, DSKRAT may be run while answering is on, only to check disk integrity, but any file being accessed may show "free" clusters.

- **SWITCHES**

**/L** Specifies that lost cluster recovery is to be performed. While the scan is being done, DSKRAT builds a file in [1,2] on the structure being processed called LSTBLK.001. If there are many lost clusters on a structure, DSKRAT will sometimes build more than one file; the second file would be LSKBLK.002, etc. The RIB of the file will point to all the clusters on the structure that DSKRAT determines are lost. After the program completes, if there are no problems reported other than lost clusters, the Supervisor may delete the LSTBLK file to recover the lost clusters. The /L switch will fail if there is no free space on the structure, or there is no room to create the RIB for the LSTBLK.??? file.

**/F** Fast processing requested. If this switch is specified then DSKRAT does

the file lookups itself instead of working through the MONITOR. This switch should be used when you are scanning a structure a second time to verify that the /L option used previously, worked correctly. (If the /L worked, then there should be no 'lost' clusters on any subsequent scan, because the clusters will be allocated to the LSKBLK.001 file.)

• **EXAMPLE:**

```
R SPT:DSKRAT
* =DKL:
% DSRIS Improper scheduling set
% DSRMOJ More than one job logged in
Running
Structure DKL   Analysis begun at  10:48:12  13-Jan-89
Number of blocks per cluster: 5
DKL0: starts at cluster 0
DKL1: starts at cluster 55746

RESPON.DAT [1,200] -- Cluster 1082  is used but not marked in SAT

The following clusters are lost (marked in use, but in no file):

004021-004060,004449,004455-004474,004905-004924,004949-
004968,004993-004998,005719-005725,005728-005733,005801-
005806,006143-006151,006183-006188,006751-006756,007168-
007984,010915-010934,010974-010993,011947-011966,012304-
012323,012458-012477,012506-012525,012866-012885,012953-
012972,015883-015902,016172-016191,016272-016291,016308-
016327,037653-037672,061942-061961,061971-061990,063093-
066876,067865-067904,068174-068193,068653-068672,068816-
068835,069829-069848,069856-069875,070359-070378,070552-
071810,071825-071844,071909-071948,072000-072019,072074-
072093,072335-072354,072537-072556,072605-072624,072668-
072687,072924-072963,073037-073056,073367-073386,073445-
074428,074452-074457,075734-075753,075819-075838,075849-
075868,076030-076049,076378-076417,076427-076446

Total number of lost clusters = 1416

The following clusters are free (not marked in use, but in some file):
001082

Total number of free clusters = 1
There are no multiply used clusters (belong to more than one file).

End of pass 1 -- No need for pass 2
OK
```

- **BATMAN**

Occasionally a region on a disk pack will become unusable. This can occur for several reasons such as dirt or scratches. The monitor will perform complete maintenance of its own bad blocks. When the monitor detects a bad disk block, it attempts to avoid reusing the block and causing future errors. This is accomplished by making an entry in the BAT block.

BATMAN is used to mark out bad spots on a disk pack. When either BATMAN or ONCE DIALOG (Reload Code B) is used to mark out bad spots, the structure must be refreshed in order for the monitor actually mark the spots as unavailable for future use. This is because the monitor only reads the BAT block during refreshes. In addition, refreshing creates the file [1,4]BADBLK.SYS, which physically contains the bad blocks. For more information on marking out BAD spots see the PROCEDURES MANUAL.

- **EXAMPLE:**

```
R SPT:BATMAN
Modify BAT blocks on what unit?  DZA2
Type HELP if you need it.
DBA2 (FDA2:602580) :  ADD
First bad blocks LBN (RETURN if no more) :
131846
Last block in region (RETURN if only 131846):
<CR>
Added 1 block at 131846
First bad blocks LBN (RETURN if no more):  <CR>
DBA2 (FDA2:602580) :  WRITE
Updated BAT blocks have been written on DZA2.
Modify BAT blocks on what unit? <CR>
OK
```

- **BATLST**

BATLST lists known bad regions (by logical block number) on a disk pack. The listing includes a "previous BAT" section so changes since the BAT was last saved will be available. If you request, the current BAT will be saved as the "previous BAT" for future BATLST runs.

## U

- Manuscript

The backup menu:      Log into the shutdown account and follow the menu:

• **EXAMPLE:**

```
System Shutdown Menu

0.    RETURN TO PREVIOUS MENU
1.    SHUTDOWN SYSTEM
2.    CANCEL SHUTDOWN

MAKE SELECTION, PRESS (ENTER): 1 <CR>

SHUTDOWN SYSTEM          (DEFAULTS are in [ ] )
SHUTDOWN WHEN? Specify when in any of the following forms:
'now' or '+nn' minutes. [+5] <CR>
REASON FOR SHUTDOWN? [(none)] <CR>
WHEN WILL THE SYSTEM BE BACK UP? [ LATER ] <CR>
REBOOT AFTER SHUTDOWN (Y/N)? [Y] <CR>
System mhaez will reboot at +5
Because of:
It will be back up later
Do you want to proceed? (Y/N) Y
Depress ENTER to Continue <CR>
(At this point you will return to the SYSTEM SHUTDOWN MENU and will be able to
wait for the system to go down or CANCEL the shut you just set.)
```

• **Windows NT Reload:**

- a). **CTRL - ALT - DEL**
- b). Select the **Shutdown** Option
- c). If you are reloading the system, select **Shutdown and Restart**  
If the system is going down for work, select **Shutdown**

• **OS/2 Reload:**

If you know the drive (partition) OS/2 is located on:

Go to the MS Dos prompt and type:      **SET BOOT /IBD: X**  
( X = the drive (partition) that OS/2 is located on.)

If the machine has a mouse:

- a) Select same channel on mouse master console.
- b) Use mouse to point to an empty area on the desktop.
- c) Click the right mouse button.
- d) Select Shut down.
- e) Select Yes in all boxes asking if a particular program is to be exited.
- f) Wait for a message that states that the shut down is complete.
- g) Reboot computer using Ctrl+Alt+Del keys or reset button.
- h) Skip to step i.

**If the machine does not have a mouse:**

- a) Press Alt+Shift+Tab.
- b) Press Ctrl+\ to deselect the icons on the desktop.
- c) Press Shift+F10 to display the pop-up menu for the desktop.
- d) Use an arrow key to highlight Shut down.
- e) Press Enter.
- f) Use arrow keys to highlight then press Enter to select Yes in all boxes asking if a particular program is to be exited.
- g) Wait for a message that states that the shut down is complete.
- h) Reboot computer using Ctrl+Alt+Del keys or reset button.
- i) After booting, the machine should run all desired programs and bring itself back up completely.

**SC RELOAD CODES**

- Normal reload code <cr>
- Common codes to use during reloads:

<b>&lt;CR&gt;</b>	Normal production system load .
<b>B</b>	Report bad regions on disk.
<b>D</b>	Do long disk dialog.
<b>H</b>	Help, types list of codes.
<b>R</b>	To refresh a structure.
<b>S</b>	Stand-alone. Do not run STARTR; log CTY into [1,2], but not through LOGIN; system load number is not updated.

- SC30's verses SC40's

- **Example of SC reload:**

```

Issue of SYSTEM KILL ALL
<Ctrl> \ (This will give you a MSP prompt.)
MSP> HALT
MSP> R BOOTSI (R BOOT40 for SC40s)
Boots %25B(154)

Checking SI0 RH1 SI2 RH3 SI4 SI5 RH6 RH7

BTS> DKE: { [1,22]MONITR.EXE, [1,22]OLDMON.EXE }

CompuServe CHE014 (built 15:12 31-May-93 in [340,100])

Why reload? <cr> Why not?
Code: <cr>

SCSI device status:
Disk:

```

RH	Bus	SCSI id	Unit	Vendor	Product Id	Rev	Serial number
0	0	0	DZA0	FUJITSU	M2263H-584	0194	J1180
0	0	1	DZA1	FUJITSU	M2263H-584	0194	J1175
0	0	2	DZA2	FUJITSU	M2263H-584	0194	J1171
0	0	3	DZA3	FUJITSU	M2263H-584	01A5	J1195
0	0	4	DZA4	FUJITSU	M2263H-584	0194	J1084

```
0 0 5 DZA5 FUJITSU M2263H-584 01A5 J1056
0 0 6 DZA6 FUJITSU M2263H-584 01A5 J1145
```

This monitor is configured for 17 controllers (16 online, could add 1).  
Tape:

RH	Bus	SCSI id	Unit	Vendor	Product Id	Rev	Units
5	0	1	MZA1	FUJITSU	M1016B	M2481B4	0010 0 1
5	0	2	MZA2	FUJITSU	M1016B	M2481B4	0010 0 1

[Reading WWVB clock.]

;;DTECRS: [1,2] job 2, 20-June-93 1:40 \*\*\*\* M E S S A G E \*\*\*\*

DTE crash file is CHA314.BIN

[It's now 1:40:32 AM EDT Wednesday, June 20, 1993 (per WWVB/Arl)]

[STRSAJ Starting autostart jobs]

[OPRPAF Processing auto command file]

:LIBRARY [1,103]

:FORCE OFF

:SLOGIN 1.2;OPCON

:DEFINE OPC=

:SLOGIN 1.2;ATOLIB

01:40:40(OPC\1)

Job 8 on CHE at 01:40 EDT 20-Jun-93 on TTY11

Class = .OPR

01:40:40(1)

Job 9 on CHE at 01:40 EDT 20-Jun-93 on TTY12

Class = .OPR

:DEFINE WORK=

:SLOGIN 1.2;ATOLIB

01:40:42(WORK\2)

MTA10: => MZA10:

MTA11: => MZA11:

MTA12: => MZA12:

MTA13: => MZA13:

01:40:45(2)

Job 10 on CHE at 01:40 EDT 20-Jun-93 on TTY13

Class = .OPR

MTA10: => MZA10:

MTA11: => MZA11:

MTA12: => MZA12:

MTA13: => MZA13:

01:40:48(WORK\2)

OK

:DEFINE OPR=

OPR-R SPT:CHKSRU

CHKSRU V1B

! Checking SRU generation at 20-Jun-93 01:40:44 EDT

! CHKSRU: Acceptable Result:

100 is within 10% of 100

OK

01:40:42(OPC\OPR)

OPCON Version 1H(1451)

Function

State

MOUNT

Active

Device(Type)

Status

Tape

Mount Count

DZA0:(M1800) Active DKE:

(20)

```
DZA1:(M1800) Active ZBD: (3)
DZA2:(M1800) Active ISW: (3)
DZA3:(M1800) Active ZAE: (3)
DZA4:(M1800) Active ZAE: (3)
DZA5:(M1800) Active ZBE: (4)
DZA6:(M1800) Active ZAF: (3)
DZA7:(M1800) Dormant

:SL 103.3;HSTCHK
01:40:46(3)
      Job 16 on THE DUBLIN EXPRESS at 01:40 EDT 20-Jun-93 on TTYP14
      Class = .INH

:W OPR
OPR 10 1,2 CHKSRU ^C 94
@[1,103]CHKMZA
SYSTEM LOG LOCAL ON
01:40:53(OPR)
      OK
      .set noecho
01:40:58(OPR)
      OK
@[1,103]LZRNRT
01:41:07(OPR)
      0141 EDT 20 Jun 1993
      OK
      OK
      .set no output
;OPSER STARTM.ATO FILE
:AUTO /<2359 AT2400.CHE[103,700]
:AUTO /<2200 AT2200.CHE[103,700]
:AUTO /<2000 AT2000.CHE[103,700]
:AUTO /<1745 AT1800.CHE[103,700]
:AUTO /<1600 AT1600.CHE[103,700]
:AUTO /<1400 AT1400.CHE[103,700]
:AUTO /<1200 AT1200.CHE[103,700]
:AUTO /<1000 AT1000.CHE[103,700]
:AUTO /<0800 AT0800.CHE[103,700]
%OPRSPR superseding previous :AUTO request
:AUTO /<0600 AT0600.CHE[103,700]
%OPRSPR superseding previous :AUTO request
:AUTO /<0400 AT0400.CHE[103,700]
%OPRSPR superseding previous :AUTO request
:AUTO /<0200 AT0200.CHE[103,700]
%OPRSPR superseding previous :AUTO request
01:41:47(OPR)
      OK
      .return 1
      OK
      .return %status
      OK
[ At this point only LOCAL answering is on]
SYSTEM LOG ALL ON
SYSTEM L
```

## **HANDLING SYSTEM CRASHES**

- How to detect a crash
- Types of crashes
- Running crash
- Box timeout crash
- Associated crash forms
- Performing crash procedures
- Who to contact when a system crashes
- DTECRS
- Node crashes

## **DISK MAINTENANCE**

### **REFRESHES**

- Done to clean up disk space
- Fragmentation and disk I/O
- Saves and verifies [\*,\*]
- Re-save and re-verify all questionable areas
- Areas not saved
- REFRESH WITH SYSTEM UP VERSES REFRESH WITH SYSTEM DOWN
- ZWIMAL

#### **Example:**

```
R SPT:ZWIMAL
Zwimal %6A(10)
Units we can read:
DHC2 <cr>
DHC3 <cr>
<cr>
Units we can write:
DHC2 <cr>
DHC3 <cr>
<cr>
Code: R
No units in active swapping list
Str to be refreshed: MBX
Do you really want to delete ALL files from MBX? Y
Str to be refreshed: <cr>
OK
```

## **RESTOR**

- Restores entire structure after a refresh has been performed
- Restores data contiguously
- **Example:**

```
R SPT:RESTOR
/ TA MTA10,MTA11,MTA12
/ U
/ G

File structure: TEST
[1,2]      [1,4]      [1,200]    [1,177777]
.
[4557,301] [65777,101]
[Restore completed]
OK
```

The following example shows a RESTOR for a move of parts of multiple structures to one structure.

```
R SPT:RESTOR
/ TA MTA10,MTA11,MTA12
/ U
/ RES TEST=[*,*]
/ RES [340,*]
/ G

File structure: TEST
Tape structure: MBX
[1,2]
[777,3]
Tape structure: DKL
[340,1]
.
[340,77777]
[Restore completed]
OK
```

## **DISK DISASTER**

- Head crash
- Other hardware problems
- Rebuild procedures
- System backups
- Papermill checklists [103,5]

## **REPORTS**

- **MORNING HOST CHECKLIST**
  - Listing is generated each morning
  - What program HSTCHK checks
- **HOST CONFIGURATION REPORT**
  - Lists each host's Monitor version, core, total jobs, disk configuration, and swapping information
  - Used when rebuilding a structure
  - Updated every Saturday at 08:00
- **OPERATIONS 24-HOUR REPORT**
  - REPORT.DAT on \\NTAADAT2\\DATA\\GROUPS\\OPR\\24HrREPORT
  - Operators make entries for any problems encountered (system crashes, hardware/software problems, etc.)

## **COMPUTER ROOM EMERGENCY PROCEDURES**

### **FIRE AND SPRINKLER SYSTEM**

- Location and function of controls
- Fire walk

### **HALON 1301 SYSTEM**

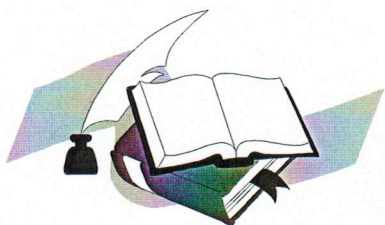
- Protected areas
- Careful of accidental dumps
- Hand held fire extinguishers

### **UPS SYSTEM**

- Function of UPS
- Display panel
- Power failure
- Backup power supplies (Battery, Diesel)

### **ROLM TELEPHONE SYSTEM**

- Paging ( **73** Dublin, **70** Arlington, **6PAGE** Hilliard)
- Transferring calls (Press **TRANSFER**, dial the **extension number**, announce the caller and then hang up)
- Picking up calls (**\*30** will pick up any extension while in the computer room)  
(**\*3** then type the number you need to pick up)
- Retrieving phone mail ( dial **39800** and **#**, then enter the **password** for that phone and **#** again. Select **3** or **LISTEN** to retrieve messages.)



# Review

## 1. System performance checking

- Getting immediate typeout of system status using SYSTAT
- Cleaning up free users disk areas using NITELY
- Creating system error reports using ERRDMP
- Morning host checklist
- Operations 24-Hour Report

## 2. Operations functions toward system performance

- Performing refreshes
- Restoring data after a refresh

## 3. Loading a system

- To load monitor from disk
- To load monitor from magtape

## 4. Reload codes

## 5. System crashes

- Detecting a crash
- Different types of crashes
- Forms to fill out
- Whom to notify
- Scheduled task to Xerox the crash forms and send them to the MONITOR group.



## Examination Seven

1. What does the program ERRDMP do?
2. Give the SYSTAT command that would give us disk performance statistics for DZA2:?
3. Explain briefly why we do refreshes?
4. What is the monitor?
5. What code would you use to reload the system for a refresh?
6. What are some types of system crashes?



# Total Review

- o Job information
- o Work schedules
- o Manuals
- o Time sheets
- o Hardware types
- o Security guidelines
- o The Monitor, MFD, UFD, and RIB
- o Customer identification
- o Online storage
- o Filename specification
- o system configuration
- o Control characters
- o ICS commands
- o Utility programs
- o OPSER
- o Logging on procedures
- o UPF files
- o LOCATE.PFE
- o Overhead programs
- o Independent processing
- o SYSDIS
- o SYSTAT
- o SYSTEM
- o OPCON
- o Magtape information
- o Disk information
- o Operator tasks
- o Magtape library system (TAPES)
- o FILSAV
- o BACKUP
- o CONMAG
- o Printer files
- o COMSPL
- o Related magtape programs
- o Billing
- o Checking system performance
- o ERRDMP
- o Morning host checklist
- o Loading the system
- o Reload codes
- o Refreshes
- o Handling system crashes

## Appendix

### OPSER

<b>:W</b>	TELLS THE STATUS OF THE SUBJOB OPSER IS POINTED TO.
<b>:W OPR</b>	POINTS TO SUBJOB "OPR" AND GIVES STATUS.
<b>:W ALL</b>	JOB STATUS OF ALL SUBJOBS, POINTER DOES NOT MOVE.
<b>OPR-</b>	POINTS TO SUBJOB "OPR", DOES NOT GIVE STATUS.
<b>WORK-CAT</b>	POINTS TO SUBJOB "WORK" AND PERFORMS THE COMMAND "CAT". DOES NOT GIVE STATUS.
<b>:LOGIN 131,1</b>	LOG A SUBJOB IN FOR [131,1]
<b>:SL</b>	SILENT LOGIN TO [1,2]
<b>:D DWN=</b>	DEFINES THE CURRENT SUBJOB AS "DWN"
<b>:D ZAB=DKB</b>	CHANGES THE SUBJOB NAME "DKB" TO "ZAB" AND MOVES OPSER POINTER TO SUBJOB "ZAB"
<b>OFF, LOG, BYE</b>	LOGS OFF A SUBJOB (MONITOR COMMAND).
<b>:Xn</b>	n = A CONTROL CHARACTER, :XC = ^C.
<b>:EXIT</b>	EXIT OUT OF OPSER IF NO SUBJOBS ARE ACTIVE.
<b>:HALF</b>	HALF DUPLEX, NO CHARACTERS WILL ECHO TO TERMINAL.
<b>:FULL</b>	FULL DUPLEX, CHARACTER TYPEOUT WILL RESUME.
<b>:A AT1400.ATO[103,700]</b>	MANUALLY STARTS AUTO JOB FOR 14:00.

### MONITOR COMMANDS

<b>BYE</b>	LOG OFF A TERMINAL OR SUBJOB.
<b>CAT</b>	CATALOG (LISTING OF UFD), SHOWS ONLY FILE NAMES.
<b>COMBINE</b>	MERGES FILES INTO ONE FILE. EX: COMBINE ONE.DAT,TWO.DAT TO THREE.DAT
<b>CONtinue</b>	CONTINUE A PROCESS AFTER A ^C INTERRUPT.
<b>COPy</b>	COPIES OR DUPLICATES A FILE. EX: COP FRA:[140,65]FILE.DAT TO DKE:[1,2]WORK.DAT
<b>CREATE</b>	CREATES A NEW FILE USING FILGE.
<b>DAYtime</b>	TYPES OUT DATE AND TIME.
<b>DELeTe</b>	DELETES FILES FROM DISK. EX: DEL NAME.EXT DEL DKE:[131,1]*.DAT
<b>DIRectory</b>	DIRECTORY OF A FILE.

#### SWITCHES:

**/A** - ACTUAL (AS OPPOSED TO ALLOCATED)  
**/B** - BLOCKS  
**/BIN** - BINARY  
**/G** - DISPLAYS A GRAND TOTAL OF STORAGE  
**/L** - LONG DIRECTORY.  
**/Q** - LISTS DISK QUOTA FOR A [P,PN] (DISK STORAGE LIMIT)  
**/SOR:OPTION** - SORT BY...  
    **EXT** - EXTENSION OF FILE.  
    **CRE** - CREATION DATE OF FILE.  
    **ACC** - LAST ACCESS DATE OF FILE.  
    **NAM** - FILE NAME.  
    **PRO** - PROTECTION CODE  
    **SIZ** - SIZE:ASC (ASCENDING ORDER) :DES (DESCENDING)  
**/T** - DISPLAYS TOTAL STORAGE FOR A [P,PN]

<b>DISmount</b>	WILL RELEASE A MAGTAPE FROM AN ASSIGNED TAPE DRIVE OR RELEASE A DISK STRUCTURE. EX: DIS MTA10: DIS SSRA:
<b>FILge</b>	CREATE OR EDIT A FILE.
<b>IND Enter FILE</b>	ENTERS INDEPENDENT JOB IN THE ICON QUEUE [3,7] FOR FILE.ICD.
<b>IND DELeTe FILE</b>	DELETES ENTRY IN ICON JOB QUEUE.
<b>IND STAtus file</b>	CHECKS THE STATUS OF INDEPENDENT JOB(S).
<b>IND KILL FILE</b>	WILL KILL AN INDEPENDENT JOB IF RUNNING OR NOT.
<b>MAG REW MTA?:</b>	WILL REWIND SPECIFIED MAGTAPE DRIVE.
<b>MAG UNL MTA?:</b>	WILL REWIND AND UNLOAD SPECIFIED MAGTAPE DRIVE.
<b>MOUnT STR:</b>	TO MOUNT THE SPECIFIED STRUCTURE (OR MAGTAPE).
<b>PAC ADD FILE.EXT TO MASTER.pak</b>	ADDS FILE.EXT TO MASTER.PAK.
<b>PAC CAT MASTER.pak</b>	CATALOG OF FILES WITHIN MASTER.PAK.
<b>PAC CLER MASTER.pak</b>	DELETES ANY DISK FILES THAT HAVE THE SAME SPECS AS FILES IN MASTER.PAK.
<b>PAC DIR/sw MASTER.pak</b>	DIRECTORY OF FILES WITHIN MASTER.PAK VALID DIRECTORY SWITCHES CAN BE USED.
<b>PAC MAKE MASTER.pak FROM FILE1.EXT, FILE2.EXT...</b>	CREATES MASTER.PAK FROM DISK FILES.
<b>PAC RETrieve FILE FROM MASTER.pak</b>	PLACES A COPY OF SPECIFIED FILE FROM MASTER.PAK TO YOUR DISK AREA.
<b>PAC PURge FILE.EXT FROM MASTER.pak</b>	DELETES SPECIFIED FILE FROM MASTER.PAK
<b>PAC TYPE FILE FROM MASTER.pak</b>	TYPE A FILE WITHIN MASTER.PAK
<b>PAC UPDate MASTER.pak</b>	OVERWRITES ANY FILES IN MASTER.PAK WITH DISK FILES THAT HAVE THE SAME SPECIFICATIONS.
<b>PRINTNH/LOC:xxx/xxM/R/xC/D FILE.EXT</b>	SPOOL A FILE TO THE PRINTER QUEUE [3,6] AND SPECIFY: PRINTER LOCATION, FORM #, PROMPT FOR ROUTING, # OF COPIES, DELETE FILE AFTER SPOOLING TO [3,6]
<b>PROtect STR:[P,PN]FILE.EXT(n)</b>	n = PROTECTION LEVEL
<b>REName STR:[P,PN]FILE1.EXT TO STR:[P,PN]FILE2.EXT</b>	CANNOT RENAME ACROSS STRUCTURES!
<b>Request (DEV:)PROGRAM</b>	PLACES CORE IMAGE FILE (PROGRAM) INTO CORE AND STARTS PROGRAM.
<b>SPOolq LCO</b>	CHECK THE PRINTER QUEUE FOR LOC:COL,FCS,INH,BIG.
<b>SPOolq/LOC:XXX</b>	CHECK THE PRINTER QUEUE FOR SPECIFIC PRINTER LOCATION.
<b>TYPE FILE</b>	TYPE OUT THE SPECIFIED FILE(S) WITHOUT A HEADING.

**SYSTEM**

**SYSTEM START** START OVERHEAD JOBS  
**SYSTEM SHUT XXXX** SHUT SYSTEM DOWN AT SPECIFIED TIME.  
**SYSTEM SHUT STR YYY XXXX** SHUT STRUCTURE YYY AT SPECIFIED TIME.  
**SYSTEM SHUT CAnceL** CANCEL A PREVIOUS SYSTEM SHUTDOWN.  
**SYSTEM LOG XXX ON (OR OFF)** TURN ANSWERING OFF (DISABLE ACCESS)  
FOR XXX, WHERE XXX CAN BE:  
**ALL** - ALL TERMINALS  
**STR XXX** - FOR DISK STRUCTRE XXX  
**DATASET** - PHONES  
**FREE** - NON-BILLABLE PPNS  
**ICON** - INDEPENDENT JOBS  
**LOCAL** - COMPUTER ROOM TERMINALS  
**REMOTE** - HARDWIRE TERMINALS  
**SYSTEM KILL X[P,PN]** KILL JOB NO. "X" FOR [P,PN].  
**SYSTEM KILL ALL** KILL ALL JOBS ON SYSTEM.  
**SYSTEM KILL HUNGUP** KILL HUNG-UP JOBS.  
**SYSTEM SYSdis (COMMAND)** CHANGE SYSTEM DISPLAY SCREEN.  
COMMANDS:  
**CLASS** - CLASS NAME FOR EACH JOB  
**DISK** - FREE DISK STORAGE  
**EMP XXX** - EMPHASIZE JOB(S):  
    **STR XXX** - FOR STRUCTURE XXX  
    **PROG XXXXXX** - PROGRAM NAME  
    **PROJ XXXXX** - PROJECT XXXXX  
**ESPrIt** - CHANGE COLOR ON ESPRIT  
**IDLE** - TIME JOB'S BEEN IDLE  
**JOB XX** - DETAILED JOB INFO.  
**PCT** - % OF PROCESSOR USED  
**PER** - PERFORMANCE STATS (%)  
**REF** - CLEAR SCREEN  
**RER** - CLEAR SCREEN AND RESET  
**SCR X** - GO TO SCREEN #"X"

**SYSTAT**

**SYS B** BUSY DEVICE STATUS  
**SYS F (=XXX)** FREE DISK STORAGE STATUS.  
**SYS O** LIST ALL PRIVATE (MOUNTABLE) STRUCTURES ONLINE.  
**SYS P (=XXX)** DISK PERFORMANCE. XXX = STRUCTURE OR DISK DRIVE.  
WILL LIST CORECTABLE, SOFT, HARD, AND RIB ERRORS.  
**SYS Q** SHORT JOB STATUS, OMITTING [1,\*] AND [2,\*].  
**SYS job#** STATUS OF SPECIFIC JOB.  
**SYS [P,PN]** SHORT STATUS OF SPECIFIC [P,PN]

### FILSAV

**INS str:[\*,\*] TO MTA?:/S**  
**CMS str:[\*,\*] TO MTA?:/S**  
**SPS str:[p,pn]FILE.EXT TO MTA?:/switch**  
**RES MTA?:[p,pn]FILE.EXT TO str:[p,pn]/switch**  
**VER MTA?:[\*,\*] WITH str:/switch**  
**DIR MTA?:[\*,\*] TO str:[p,pn]file.dir**

**SWITCHES:**   **/D** - REGARDLESS OF DATE (RESTORE)  
                 **/F** - LIST FILE NAME(S)  
                 **/I** - IGNORE FILES ON DISK, BUT NOT ON TAPE DURING VERIFY.  
                 **/L** - LONG FILE DIRECTORY  
                 **/M** - CHANGES FILE CREATION DATE FOR MARKETING  
                      PPNS [N777,\*] SO NITELY WON'T DELETE THEM  
                 **/P** - CREATE DIRECTORY OF TAPE WHILE SAVING.  
                      FILE WILL BE CALLED XXX.DIR, XXX=STRUCTURE  
                 **/S** - SUPPRESS TYPEOUT OF PPNS.  
                 **/U** - DO NOT UNLOAD TAPE.

### BACKUP

**TAPE MTA?**

**FILES**

**DEN 6250**

**SSN XXXXXX**

**EOT**

**SKIP (-)n**

**REWIND**

SAVE SET NAME ("ALL" IF RESTORING ENTIRE TAPE)  
SKIPS TO END OF TAPE (FOR APPEND)  
SKIPS n FILES FORWARD OR BACKWARD (+ OR -)  
REWIND TO BEGINNING OF TAPE

**SAV STR:[P,PN]FILE1.EXT,STR:[P,PN]FILE2.EXT**

**RES STR:[P,PN]OUTPUT.EXT = STR:[P,PN]INPUT.EXT**

**NPRI STR:[P,PN]FILE.DIR** (CREATE A DIRECTORY FILE OF WHAT'S ON TAPE)

**CHECK STR:[P,PN]\*.\*** (VERFIY THAT TAPE IS SAME AS DISK)

### CONMAG

**DISK TO TAPE:**

**STR:[P,PN]FILE.EXT TO MTA?:/MOD:XXX/PRE:###/BLO:###/DEN:####/LAB:XXX <ESC>**

**/MOD:**           EBC = EBCDIC           USA = USA ASCII  
**/PRE:**           PHYSICAL RECORD LENGTH  
**/BLO:**           BLOCKING FACTOR  
**/DEN:**           DENSITY OF TAPE  
**/LAB:**           NON = NONE (NO LABEL)    IBM = CREATE IBM LABEL

**TAPE TO DISK:**

**MTA?:/MOD:XXX/PRE:###/BLO:### TO STR:[P,PN]FILE.EXT/LIN:X <ESC>**

**/LIN:X**           TYPE OUT X NUMBER OF LINES  
**/ADV:X**           ADVANCE X FILES (TO SKIP OVER A HEADER FILE)

## BILLING

@BILLING is run automatically from the 18:00 Auto job. Can be run manually for a non-normal run.

**R OPLDAY** (IF CIS STRUCTURE IS ON THAT SYSTEM)  
**@BILSAV n** TO SAVE BILLING .PAK FILES TO MTAn:  
**@BILSAV n,RES,str** TO RESTORE BILLING .PAK FILES TO A STRUCTURE  
(SEE SOP264 FOR THE CONTINUATION OF BILLING IN ARLINGTON).

## NTCOPY

YOU WILL BE PROMPTED FOR REMOTE HOST, USER ID, AND PASSWORD (YOU MUST KNOW THE PASSWORD!) THEN YOU MAY SEND A FILE TO THE REMOTE PPN, OR RECEIVE A FILE FROM THE REMOTE PPN:

- \* **SEN FILE.EXT**
- \* **REC FILE.EXT**

IF YOU ARE LOGGED INTO [1,2] YOU MAY:

- \* **SEN STR:[P,PN]FILE.EXT**
- \* **REC FILE.EXT TO STR:[P,PN]**

- \* **CAT** CATALOG FILES IN REMOTE PPN.
- \* **DEL** DELETE SPECIFIED FILE IN REMOTE PPN.
- \* **EXIT** EXIT FROM PROGRAM.

\*\*\* OR \*\*\*

**COPY STR:[P,PN]FILE.EXT TO STR:[P,PN]\*.\* /R**

**COPY STR:[P,PN]\*.\* FROM STR:[P,PN]FILE.EXT /R**

YOU WILL BE PROMPTED FOR PASSWORD AND ACCOUNT I.D.(IF ANY).  
COPY/REMOTE IS FASTER THAN NTCOPY!

**TAPES                      RUN FROM [100,4])**

<u>MAGLIB</u>		<u>FILSAV</u>	
T	TEMP. - KEEP AT MOST 1 MONTH	D	DAILY - KEPT 7 DAYS
R	REGULAR - KEPT UNTIL EXP. DATE	W	WEEKLY - KEPT 8 WEEKS
P	PERMANENT - KEPT FOREVER	M	MONTHLY - KEPT 1 YEAR
		B	BOND - KEPT 45 DAYS
C	CARTRIDGE (Cartridge and DAT tapes)		
K	CANISTER (Tapes in canisters - rare)		
H	HANGING (reel to reel)		

**INS M**                      INSERT A TAPE INTO MAGTAPE LIBRARY (WILL PROMPT)  
**INS F**                      INSERT FILSAV TAPES (WILL PROMPT FOR STR AND PPNS)

**FIN M[P,PN]'TAPEID'** \*\*\***WARNING**\*\*\* DO NOT ENTER FIND M<CR>, TYPE IN  
THE ENTIRE COMMAND LINE OR YOU WILL GET A LISTING OF THE  
ENTIRE LIBRARY. (:XP MAY STOP THE TYPEOUT).

**FIN F**                      FIND SLOT NUMBER OF A FILSAV TAPE (WILL PROMPT)

**ALL F/STR:XXX** FIND ALL FILSAV TAPES FOR THAT STRUCTURE.

**ALL F/DA:XX/XX** FIND ALL FILSAV TAPES FOR THAT DATE.

**REL M** RELEASE A TAPE FROM MAG. LIBRARY (WILL PROMPT)

**REL M#xxxxxx** RELEASE TAPE FROM SPECIFIC SLOT NUMBER (FASTER!)

**PULL**                      RELEASE EXPIRED FILSAV TAPES, DONE AT 12:01 DAILY.

**SEND**                      KEEPS TRACK OF TAPES SENT OUT OF COMPUSERVE.

**MOU M#xxxxxx**                      TO UPDATE THE LAST DATE THE TAPE WAS MOUNTED.

**TASKS                      (RUN FROM [100,4])**

**Sxxxx**                      TASK OFF SCHEDULED TASK # xxxx.

**Fxxxxx**                      TASK OFF FORM # xxxxx.

**CHA Fxxxxxx**                      ALLOWS CHANGES TO BE MADE TO A TASK COMPLETED WITHIN THE LAST 6  
HOURS. SCHEDULED TASKS MAY ALSO BE CHANGED.

**CHE Fxxxxxx**                      LISTS COMPLETED TASK (OR RANGE OF TASKS).

**L [P,PN]**                      LISTS COMPLETED TASKS THAT MATCH THE SPECIFIED PPN.

**P n**                              LISTS PENDING TASKS FOR THE NEXT 'n' HOURS.

## OPCON

OPCON PROCESSES ALL REQUESTS TO MOUNT OR DISMOUNT MAGTAPES OR DISK PACKS.

MESSAGE ON CTY: "# MOUNT DISK STRUCTURE (OR MAGTAPE) xxxxx ON DEV:"  
[PHYSICALLY MOUNT THE TAPE OR DISK PACK]

### **OPC-# (ON DEV:)**

OPCON WILL THEN TRY TO PROCESS THE COMMAND AND TELL OPERATIONS WHETHER THE MOUNT WAS SUCCESSFUL OR NOT.

OTHER COMMANDS:

<b>RUL STR:</b>	REQUEST TO UNLOAD A STRUCTURE. ***MAKE SURE OPCON SAYS "STRUCTURE HAS BEEN UNLOADED" BEFORE REMOVING THE DISK PACK!***
<b>KILL XX</b>	CANCELS REQUEST NUMBER XX, WILL ASK FOR REASON
<b>PEEK</b>	TYPES PENDING OPCON REQUESTS.
<b>STA D</b>	TYPES A STATUS OF DEVICES (DISK DRIVES).

## PRINTER PROGRAMS

**SPO/LOC:XXX** CHECKS PRINTER QUEUE [3,6] FOR LISTINGS WAITING TO BE PRINTED.

TAPSPL USED TO SAVE FILES FROM PRINTER QUEUE TO MAGTAPE AND RESTORE TO ANOTHER SYSTEM.

<b>* SAV LCO</b>	SAVE ALL PRINTER FILES THAT DATA CONTROL CAN PRINT IN ARLINGTON.
<b>* SAV LCO/FORM:XX</b>	SAVE FILES TO BE PRINTED ON THE SPECIFIED FORM.
<b>* SAV COL</b>	SAVE ONLY LOCATION "COL".
<b>* RES</b>	RESTORE ENTIRE MAGTAPE TO A PRINTER QUEUE.

FILL OUT LABEL!

**COMSPL**

USED TO SAVE MICROFICHE FILES FROM THE PRINTER QUEUE TO DEGAUSSED TAPE (WITH GREEN TAPE SEAL). INDEXED FILES (FORM 44) MUST BE SAVED ON A SEPARATE TAPE FROM NON-INDEXED FILES (ALL ELSE) COMSPL WILL INFORM YOU IF THERE ARE FORM 44'S.

**OPTIONS:**

**Overwrite**  
**Append**

START TO SAVE (OVERWRITE) MICROFICHE TO TAPE.  
APPENDS FILES TO THE END OF EXISTING TAPE.

FILL OUT LABEL AND SPECIAL MICROFICHE FORM!

**UTILITY PROGRAMS**

**IHMAIN**

FIND ISK	TELLS WHAT HOST ISK IS MOUNTED ON.
STR ISK	" " " " " " "
STR IS?	TELLS WHAT HOSTS ALL IS? STRUCTURES ARE MOUNTED ON.
HOST EHA	TELLS WHAT STRUCTURES ARE MOUNTED ON SYSTEM EHA.

**EMDUTL**

FIND FRA	TELLS WHAT HOST FRA IS MOUNTED ON.
FIND 140,65	TELLS WHAT STRUCTURE AND HOST THE PPN IS LOCATED ON.

INDQ

WILL CHECK THE INDEPENDENT PROCESSING QUEUE FOR  
ICON JOBS OF SPECIFIC STRUCTURE AND/OR PROJECT.

REACT

REACT IS USED TO CREATE AND MAINTAIN USER PARAMETER FILES  
([1,201]#777777,,(PROJ).UPF), WHICH CONTAIN ALL PRIVILEGES,  
RESTRICTIONS, AND PARAMETERS SPECIFYING WHAT A USER IN A  
PARTICULAR PROJECT ON A SPECIFIC SYSTEM CAN DO.

- \* **SEL <PROJECT>**                      SELECTS A PROJECT
- \* **TYP 0**                                  TYPES DEFAULT SETTINGS FOR  
   THE ENTIRE PROJECT.
- \* **TYPE <PROG. NUM.>** TYPES NAME, ADDRESS, ROUTING  
   PRIVILEGES, AND RESTRICTIONS.
- \* **EXIT**

YOYOED

(ON DEFAULT SYSTEM FOR [1,201])

USED TO MODIFY OR READ [1,201]LOCATE.PFE, WHICH CONTAINS  
A LIST OF ALL VALID PROJECTS AND THEIR DEFAULT STRUCTURES.  
YOYOED WILL ASK YOU TO ENTER A DATE OR <CR> FOR THE CURRENT  
DATE. THUS YOU CAN LOOK UP WHAT STRUCTURE A PROJECT WAS  
LOCATED ON IN THE PAST. (PAST DATE MUST BE A WEDNESDAY).  
LOCATION HISTORY FILES ARE KEPT FOR ABOUT A YEAR.

- \* **T<PROJECT>**                      TELLS THE DEFAULT STRUCTURE  
   OF A PROJECT.

NOTICE

USED TO INFORM A CUSTOMER REGARDING A REQUEST

NOTICE WILL CREATE A FILE IN WHICH YOU MUST  
TYPE THE MESSAGE FOR THE CUSTOMER. YOU  
WILL THEN BE PROMPTED FOR STRUCTURE AND PPN.

## **THE BASICS OF BSDI**

### **INTRODUCTION TO BSDI**

This document will provide a general overview on the concepts and functionality of the BSDi operating system and commands. All of the commands can be expanded upon by using the 'man' (online reference manual) command and then the command the user would like more information on.

BSDi respectively like other operating systems needs a method of keeping track of information. This is done by the use of FILES and DIRECTORIES. A FILE is collection of information stored electronically. A DIRECTORY is a file that is set aside for the storage location of other files. The arrangement of the directories is accomplished by using a HIERARCHICAL file system. The top of which is called the ROOT DIRECTORY (root). Then there are a series of primary directories extending from the root directory. Within each primary directory there may also be subdirectories and so on. The directory in which a given directory resides is called a PARENT DIRECTORY. The parent directory is portrayed by two periods (..). The current directory in which you are pointed to is portrayed by a single period (.). The root directory is identified by a single forward slash (/).

It is important to remember when working with BSDi that it is very case sensitive. Any command that is issued will be in lower case. Some commands will have many modifiers that can be upper and lower case. Be careful, one command may have both a upper and lower case modifier of the same letter, but they will mean two totally different things. This will become clearer as you become more familiar with the BSDi operating system.

### **FILE PERMISSIONS**

The BSDi file system has a unique way of protecting its files, called FILE PERMISSIONS. File permissions are broken down into three groups; owner, group, and other. With in these three groups there are three protection levels; Read, Write, and Execute.

The notations for file permissions are:

<b>r</b>	Read permission granted
<b>w</b>	Write permission granted
<b>e</b>	Execute permission granted

- Permission denied for any of the three groups.

The permissions are listed in the following manner:

**wxrwxrwx**

The first three characters denote the owner of the file, the second three the owners group, and the last three any others. As one may well guess the system administrator or persons who can log in as root have access to all of the files on the system without regard to the files permission.

## **FILE TYPES**

We have already covered a little about the BSDi file system and its use of files and directories. To take it one step further, we need to look at the explanation of the different types of files. There are three types of files on a BSDi system. These include *ORDINARY FILES*, *SPECIAL FILES*, and *DIRECTORY FILES*.

### **ORDINARY FILES**

Contain text and code for a program or document.

### **SPECIAL FILES**

Represent a physical device such as a terminal, disk drive, tape drive, or a communications link. The system reads and writes to special files the same way as ordinary files, but the request does not activate the normal file mechanism. Instead they activate the device handler associated with the device.

### **DIRECTORY FILES**

Represent a collection of files and other directories/subdirectories used to group similar types of information together in one area.

## **FILE NAMES**

A file name denotes a file within a directory on a BSDi files system. Typically the files name contains up to 14 characters on a System V release. However, some different versions derived from Berkley can have as many as 255 characters.

You can use any character besides a '/' in a file name. However, some characters such as '-' or a blank space have special meanings on a command line and should not be used. Upper and lower case characters can also be used to name a file. Because BSDi is case sensitive the file names 'PROGRAM', 'Program', and 'program' will all have different meanings. Using a '.' at the beginning of a file usually denotes a initialization file or a file to be used in conjunction with a particular program.

## **PATH NAMES**

Every file on a BSDi System has a specific location. In order to use a file you must be able to locate where it is within the file system. Each file has an ABSOLUTE PATH NAME. The Absolute path name of a file is a representation of the files location from the beginning of the file system. This location is also called root. The Absolute path name is a road map to the file in question.

**/etc/passwd**

For example, the above passwd is the file name we want to look at and etc is the directory in which it is located. The etc directory is located within the root directory. So /etc/passwd is the ABSOLUTE PATH NAME for the file passwd. The passwd file is a simple one though. With the directory structure being hierarchical the absolute path name could become quite long. For this reason a file also has what is called a RELATIVE PATH NAME. A Relative path name is a file name relative to your current directory. A relative path name does not start with a /. If our current directory was /home/bshell and we wanted a look at a file called BSDi.txt within a directory called docs from bshell we could use the following relative path name.

**docs/BSDi.txt**    The Absolute path name for this file would be:    **/home/bshell/docs/BSDi.txt**

## **LOGGING IN ON A BSDi SYSTEM**

Every user has a login name and password which are assigned by the system administrator. The password should be changed once the user logs on for the first time. A password must have at least 6 characters and cannot use the # or the @ symbols. The # and @ symbols are standard error correction characters. The password should have 2 numbers and 1 special character along with the alpha characters. Passwords cannot be the same as, a reverse of, or a circular shift of the user name. When changing the password the new password must have at least 3 different characters from the previous password. Passwords are stored in /etc/passwd . The password file looks similar to the following:

login name:encrypted password:user id: group id: comment field: login home directory: program field

That is a typical record in the password file. Anyone can view the password file but only System Administrator and persons logged in as root can make changes to the file.

After the user has entered the login name and password, the system checks these with those in the password file. If the user has gained access, it establishes a user and group id, then places the user in his/her login directory, then executes the

specified program if any. The program usually is some type of shell program. If a shell is specified a prompt is displayed letting the user know the system is now ready for commands.

## **SHELLS**

SHELL is a program that controls your interaction with the BSDi system. A Shell handles the following tasks:

- \* Input / Output redirection
- \* Pipelines
- \* Metacharacters
- \* Variable assignment
- \* Conditional operators
- \* Command Substitution

The shell is a command interpreter and a powerful programming language. The shells are interactive and acts as a liaison between you and the operating system. The three most common Shell programs are:

- \* The Bourne Shell
- \* The C Shell
- \* The Korn Shell

## **INPUT / OUTPUT REDIRECTION**

All shell operate under what is called Standard I/O. All input comes from the keyboard and all output goes to the monitor. The symbols '<' and '>' are used to redirect input and output. For example if we wanted to mail a text file to a particular user you could use the following command:

**mail username < filename**

This will send the contents of the file on the right side of the '<' to the user specified on the left side.

Output redirection on the other hand allows the user to capture the results of a command in a file. For example the 'ls' command (displays files in your current directory).

**ls > file name**

If the file name used did not exist the system would create it, but if the file already existed the system would have overwritten the old one. The file specified would now contain a listing of all the files within the current directory.

## **PIPE LINES**

A pipe line connects the output of one command to the input of the next command. The pipe symbol (|) is used to connect processes in a pipeline. For example if you wanted a sorted list of the files within your directory you could type the following:

**ls | sort**

This would display a sorted list of files to your display screen. To combine I/O redirection and pipelines an example would be:

**ls | sort >sorted.lst**

The above command will execute the 'ls' command and then pipe the output through the 'sort' command, and then instead of putting the output to the display screen it will be placed into a file called sorted.lst. It is important to remember that I/O redirection is used when going from a command to a file, and a pipe is used when going from a command to a command.

## **METACHARACTERS (Special Characters)**

Metacharacters are a shorthand method of specifying a single or group of files. There are three different metacharacters or wild cards. These are '\*', '?', and '[]'. The operating system replaces the Symbols with the corresponding characters. The Metacharacters have the following meaning:

- \* = All
- ? = Matches any single character
- [] = Class of characters

For example, if you wanted to list the files in your directory with the names of 'shell1', 'shell2', and 'shell3' you could use this command.

**ls shell[1-3]**

This can also be accomplished with alpha characters as well. The file names this time are 'shella', 'shellb', and 'shellc'.

**ls shell[a,b,c]**

The metacharacter '?' works just like the '\*' but instead of matching all it will match any single character.

**ls shell?**

The above would list any file in the current directory that was 6 characters long and the first 5 characters were 'shell'

## **KERNEL**

The BSDi Kernel is the heart of the entire system. The **Kernel** controls all the functions that are required by the system. Processor allocation, file manipulation, memory management, and job scheduling are all controlled by the Kernel.

## **PROCESSES**

On a BSDi system there are many jobs running concurrently called processes. The main process is the kernel. Each time a user logs in or a job starts on a BSDi system the Kernel process generates a child process. Each user may also generate other processes from an existing process. These processes are called **PARENT** and **CHILD** processes. Thus, every process on a BSDi system has a Parent process except for the Kernel. There are some Processes on a BSDi system that always exist called **DAEMONS**. Daemons are processes that handle general system activities such as billing information, electronic mail, and printer queues.

As said before the BSDi operating system is able to run multiple processes at the same time. With this the user needs a way to control which process he/she is working with. Each child process created is given a process number. This number starts with the number one and increments by one for each child process generated. This is where job control comes into the picture. There are two types of processes a user can start. The first is called **FOREGROUND JOBS** (fg) and the second **BACKGROUND JOBS** (bg). The 'ps' (process status) command is used to check on the status of such jobs. The process a user is currently working in is the foreground job. If the user knew that a particular job was going to take a long time to run he/she could put the job in the background. Putting a job in the background is accomplished by putting an '&' after the command. Thus the job is started in the background and issued a job number of one, if it is the first job started.

With multiple jobs running, the user now needs a way to be able to interchange between the different jobs. This is done by using the information obtained from the 'jobs' command. The number in the left column is the job number that was assigned to that process. If the user wanted to move a process that was running in the background to the foreground the following command would be used:

**fg %[jobnumber]**

Now we have that particular job running in the foreground. With a job running in the foreground we cannot check on or do anything else unless we return it to the background. In order to do this we need to suspend the job momentarily. This is done by typing the command:

**^z (control z)**

The ^z is standard on a lot of systems but may vary depending on terminal settings. Once the user has typed the ^z a job number is reassigned and will be displayed to the left of the word suspended. To put the job in the background and continue processing you would type:

**bg %[job number]**

Processing is then resumed in the background. This use of job control allows the user to be working on multiple tasks at any one given time.

## **SYSTEM SHUT DOWN**

BSDi uses a buffering system. What this means is that when a user is modifying a file or a program is updating information it may not be writing it straight to the system disks. Instead it updates the files in memory. This buffering system save the system disks from being accessed unnecessarily. When the user/program is finished with the file the system then updates all the changes made to the disk. With buffering taking place the system disks need to be updated before the system is reset or powered off for maintenance. The 'shutdown' command is used to accomplish this. 'shutdown' provides an automated procedure to update and close all the system and user files. There are a number of options you can use with the shutdown command to do the following:

### **shutdown -option time [message]**

- f** Arranges for the filesystem not to be checked on reboot.
- h** The system is halted at the specified time.
- k** Kicks everyone off and prevents anybody else from logging on.
- n** Prevent the normal sync before stopping.
- r** Shutdown executes reboot at specified time.

The time argument is the time shutdown will take place. Time may be in the form of the word 'now' to perform the shutdown immediately or '+xx' where xx is equal to the number of minutes before shutdown is to occur, also you may specify the time by entering :

#### **yymmddhhmm**

**yy** = Year  
**mm** = Month  
**dd** = Day  
**hh** = Hour  
**mm** = Minute

## **COMMANDS**

*Due to the variations in command structure and options between versions and distributors. Users should consult the man pages for additional information on each individual command desired. The standard format for looking at the online reference manual is.*

### **'man' 'command desired'**

**cat** Reads files sequentially, writing them to standard output.

cat filename

**cmp** Compares two files of any type and writes the results to standard output.

cmp -options file1 file2

<b>cp</b>	Copy files, Makes a duplicate copy of a specified file or files from one location to another.  cp source destination
<b>date</b>	Displays or sets the date and time.  date +format
<b>df</b>	Displays free disk space.  df -options
<b>diff</b>	The diff command is a differential file comparer. It compares pairs of files and displays the differences to standard output. It can also be used to compare two complete directories.  diff -options file1 file2  diff -options path1 path2
<b>du</b>	Displays disk usage reports for a file system.  du -options
<b>file</b>	Determines the file type to standard output.  file -options file
<b>find</b>	Looks through a specified file hierarchy in search of a specified file.  find path file
<b>finger</b>	Displays information about a user currently logged in.  finger -options username/location
<b>grep</b>	Searches through a set of files for all lines that match a specified regular expression. (Global Regular Expression Print)  grep -options pattern file
<b>head</b>	Displays the specified number of lines of a file starting from the top and descending.  head -(#of lines) file
<b>kill</b>	Terminate or signal a process.  kill -options pid (Process Identification #)

- ls** Lists the contents of a directory.
- `ls -options path/filename`
- pwd** Prints the absolute path name of the current working directory.
- `pwd`
- rm** Removes files from file system. (Warning: Unlike some Operating systems BSDi does not have an undelete utility. Once a file is deleted the only recovery options is from a backup archive.)
- `rm -options path/file`
- rmdir** Removes the directory specified, provided it is empty.
- `rm -options directory`
- rwho** Who is logged in on the local system.
- `rwho -options`
- shutdown** Provides an automated system of shutting down the system for maintenance or reload.
- `shutdown -options time message`
- (NOTE: See the 'man' pages or the section in this booklet for more information concerning shutdown)
- sort** Sort or merge text files.
- `sort -options file`
- tail** Displays the specified number of lines from the bottom of the file.
- `tail -(#of lines) file`
- tar** Utility used to read and write information to magnetic tape. You can also use other disk drives and directories as a medium to output.
- `tar -option [argument] device files`
- who** Displays who is logged in.
- `who options`

## **GLOSSARY**

*This Glossary contains some terms and information that may not be otherwise mentioned in the booklet at this time. This information was thought to be important and might be included at a later date.*

<b>Archive</b>	<i>A file containing a group of files for backup or transfer purposes.</i>
<b>Buffer</b>	<i>A portion of a computer systems memory where data is stored before it is written to the system disk.</i>
<b>Cache</b>	<i>A group of buffers used to store data after it is read from a block device so that the data does not need to be reread from the system disks.</i>
<b>Client</b>	<i>A terminal/workstation connected to a server.</i>
<b>Core Dump</b>	<i>A listing of the state of the computer systems memory and other features that might indicate the cause of a malfunction or system error.</i>
<b>Daemon</b>	<i>A process that resides on a computer system continuously to gather information for particular reasons.</i>
<b>Device Driver</b>	<i>A program associated with special files to handle the performance of attached hardware.</i>
<b>Gateway</b>	<i>A computer system that is connected to two or more other networks which handles network traffic between the other networks.</i>
<b>Group</b>	<i>A set of users with a common interest recognized by the file server.</i>
<b>Host</b>	<i>A computer system which is connected to a network, using TCP/IP protocol. A computer that services requests and performs tasks of other computers.</i>
<b>Initialization file</b>	<i>A file that contains information for a specific program. That is executed to set up default values or re-occurring information.</i>
<b>IP address</b>	<i>A set of four numbers that specifies the location of a host or other computers on a network.</i>
<b>Kernel</b>	<i>The program that handles the functions of disk access, file manipulation, processor allocation, job control, as well as various other activities of the computer system.</i>
<b>Packet</b>	<i>A segment of information sent over a network. (Ethernet/Internet)</i>
<b>TCP/IP</b>	<i>Transmission Control Protocol / Internet Protocol, A communication protocol developed at Berkley that is standard for most versions of BSDi and some other networks as well. Used commonly for information exchange over the Internet.</i>